Fundamenta Informaticae 152 (2017) 51–86
DOI 10.3233/FI-2017-1512
IOS Press

Jumping Scattered Context Grammars

Alexander Meduna*, **Ondřej Soukup**
Brno University of Technology, Faculty of Information Technology
Centre of Excellence, Božetěchova 1/2, 612 66 Brno, Czech Republic
meduna@fit.vutbr.cz, isoukup@fit.vutbr.cz

Abstract. Conceptually, jumping scattered context grammars coincide with their standard counterparts, but they work differently. Indeed, a jumping version can apply a rule of the form $(A_1, A_2, \ldots, A_n) \to (x_1, x_2, \ldots, x_n)$ so it simultaneously erases $A_1, A_2, \ldots, A_n$ in the current sentential form while inserting $x_1, x_2, \ldots, x_n$ possibly at different positions than the erased nonterminals. In fact, this paper introduces and studies scattered context grammars working under nine different jumping derivation modes, all of which give rise to the computational completeness. Indeed, the paper characterize the family of recursively enumerable languages by scattered context grammars working under any of these jumping modes. In its conclusion, the paper sketches application perspectives and formulates several open problems.

Keywords: scattered context grammars, jumping derivation modes, generative power, computational completeness

1. Introduction

First, this introductory section explains the reason for introducing jumping scattered context grammars. Then, it informally describes nine jumping derivation modes in terms of these grammars. In addition, it sketches their expected application areas. Finally, it described how the paper is organized.

At present, processing information in a largely discontinuous way represents a common computational phenomenon today [1, 2, 3]. Consider a process that works with information in this way. During

*Address for correspondence: Brno University of Technology, Faculty of Information Technology, Centre of Excellence, Božetěchova 1/2, 612 66 Brno, Czech Republic.

Received October 2014; revised December 2016.

a single computational step, the process usually reads a fragment of the information, erase it, generate a new piece of information, and insert this newly generated piece into the entire information possibly far away from the original occurrence of the erased fragment. Therefore, intuitively speaking, during its computation, the process keeps jumping across the information as a whole. To explore computation like this mathematically, informatics needs formal models that reflect it in an adequate way.

Traditionally, formal language theory has always provided computer science with rewriting systems, which work on words and define languages. In this way, these systems explore various information processors strictly mathematically, so they should formalize the above-sketched information processing, too. However, the classical versions of these systems, such as automata and grammars, work on words so they erase and insert subwords at the same position, hence they necessarily fail to serve as appropriate rewriting systems for this purpose. Consequently, a proper formalization of processors that work in the way described above needs an adaptation of some classical well-known grammars so they reflect the above-described computation more adequately. Of course, simultaneously, any adaptation of this kind should conceptually maintain the original structure of these models as much as possible so computer science can quite naturally base its investigation upon these newly adapted grammatical models by analogy with the standard approach based upon their classical versions. To put it simply, while keeping their conceptualization unchanged, these formal models should work on words in newly introduced ways, which more properly reflect the above-mentioned modern computation. Therefore, in this way, formal language theory has recently introduced jumping versions of automata and grammars, such as jumping finite automata (see [4]) and jumping grammars (see [5]), which formalize the computation sketched above adequately. The present paper continues with this new and vivid topic in terms of scattered context grammars (see [6]).

To give an insight into the key motivation and reason for this study, let us take a closer look at a more specific kind of information processing in a discontinuous way. Consider a process p that deals with information i. Typically, during a single computational step, p (i) reads n pieces of information, x_1 through x_n, in i, (ii) erases them, (iii) generate n new pieces of information, y_1 through y_n, and (iv) inserts them into i possibly at different positions than the original occurrence of x_1 through x_n, which was erased. To explore computation like this systematically and rigorously, the present paper introduces and discusses jumping versions of scattered context grammars (see [7]), which represent suitable grammatical models of computation like this.

To see this suitability, recall that the notion of a scattered context grammar G represents a language-generating rewriting system based upon an alphabet of symbols and a finite set of rules. The alphabet of symbols is divided into two disjoint subalphabets—the alphabet of terminal symbols and the alphabet of nonterminal symbols. In G, a rule r is of the form

$$(A_1, A_2, \ldots, A_n) \ \rightarrow \ (x_1, x_2, \ldots, x_n),$$

for some positive integer n. On the left-hand side of r, the As are nonterminals. On the right-hand side, the xs are strings. G can apply r to any string u of the form

$$u = u_0 A_1 u_1 \ldots u_{n-1} A_n u_n$$

where us are any strings. Notice that A_1 through A_n are scattered throughout u, but they occur in the order prescribed by the left-hand side of r. In essence, G applies r to u so

(a) it deletes A_1, A_2, ..., A_n in u, after which
(b) it inserts x_1, x_2, ..., x_n into the string resulting from the deletion (a).

By this application, G makes a derivation step from u to a string v of the form

$$v = v_0 x_1 v_1 \dots v_{n-1} x_n v_n$$

Notice that x_1, x_2, ..., x_n are inserted in the order prescribed by the right-hand side of r. However, they are inserted in a scattered way—that is, in between the inserted xs, some substrings vs occur.

To formalize the above-described computation, consisting of phases (i) through (iv), the present paper introduces and studies several jumping derivation modes in scattered context grammars. However, a complete variety of all possible modes is so enormous that its exhaustive coverage is ruled out in a single paper. Therefore, we narrow our attention only to a restricted variety of these modes. More precisely, all jumping derivation modes under consideration in this paper simultaneously satisfy requirements (A) and (B), given next.

(A) In essence, this restriction requires that the simultaneous jumps never cross over each other. To explain this requirement more precisely, reconsider $(A_1, A_2, \dots, A_n) \rightarrow (x_1, x_2, \dots, x_n)$ as a rule r in G (see above). In all the jumping modes discussed in this paper, the insertion of each x_i occurs somewhere in between the insertion of x_{i-1} and that of x_{i+1}, so G inserts the strings x_1, $x_2, \dots, x_n$ in this order into the produced string. Consequently, G never inserts any x_i somewhere in front of x_{i-1} or behind x_{i+1}.

(B) G inserts x_1 and x_n in the way described either in (B1) or in (B2).

- (B1) G inserts x_1 somewhere behind the original position of A_1, and simultaneously, it inserts x_n somewhere in front of the position of A_n.
- (B2) G inserts x_1 somewhere in front of the position of A_1, and simultaneously, it inserts x_n somewhere behind the position of A_n.

Under restrictions (A) and (B), we next classify the jumping modes discussed in this paper into the following four types—I through IV, some of which consist of several modes, however. In total, the paper includes nine modes, Modes 1 through 9, described below in a greater detail.

I. As obvious, the standard derivation step, sketched above, can be seen as a special jumping mode. Indeed, this mode actually requires that G applies $(A_1, A_2, \dots, A_n) \rightarrow (x_1, x_2, \dots, x_n)$ in such a way that during the n simultaneous jumps, each x_i is placed exactly at the position of the deleted A_i (Mode 1).

II. G satisfies restrictions (A) and (B); otherwise, it places no other restriction on the n simultaneous jumps. As a result, within this type, (B1) and (B2) give rise to Modes 2 and 3, respectively.

III. In essence, G makes the n simultaneous jumps so each of them jumps exactly at the positions of the neighbouring rewritten nonterminals. As a matter of fact, this type gives rise to four specific jumping modes, Modes 4 through 7, described below in a greater detail.

IV. G makes each jump anywhere, but its distance is limited by the rewritten neighbouring nonterminals. By analogy with II, (B1) and (B2) give rise to Modes 8 and 9, respectively.

Of course, even under restrictions (A) and (B), these nine modes should not be considered as their exhaustive list because as obvious, some of them might be further modified. In the authors' opinion, however, they cover the essential variants of jumping modes under restrictions (A) and (B) in terms of scattered context grammar. (In the conclusion of this paper, as an inspiration for the future investigation, we briefly suggest more jumping modes, including a mode that drops (A) and (B); in fact, some of them can make the simultaneous jumps that cross over each other in either direction.)

Next, we sketch the nine modes in a greater detail.

(1) Mode 1 requires that $u_i = v_i$ for all $i = 0, \dots, n$ in the above described derivation step.

(2) Mode 2 obtains v from u as follows:

(a) $A_1, A_2, \dots, A_n$ are deleted;
(b) x_1 through x_n are inserted in between u_0 and u_n.

(3) Mode 3 obtains v from u so it changes u by performing (3a) through (3c), described next:

(a) $A_1, A_2, \dots, A_n$ are deleted;
(b) x_1 and x_n are inserted into u_0 and u_n, respectively;
(c) x_2 through x_{n-1} are inserted in between the newly inserted x_1 and x_n.

(4) In mode 4, the derivation from u to v is performed by the following steps:

(a) $A_1, A_2, \dots, A_n$ are deleted;
(b) a central u_i is nondeterministically chosen, for some $0 \le i \le n$;
(c) x_i and x_{i+1} are inserted into u_i;
(d) x_j is inserted between u_j and u_{j+1}, for all $j < i$;
(e) x_k is inserted between u_{k-2} and u_{k-1}, for all $k > i + 1$.

(5) In mode 5, v is obtained from u by (5a) through (5e), given next:

(a) $A_1, A_2, \dots, A_n$ are deleted;
(b) a central u_i is nondeterministically chosen, for some $0 \le i \le n$;
(c) x_1 and x_n are inserted into u_0 and u_n, respectively;
(d) x_j is inserted between u_{j-2} and u_{j-1}, for all $1 < j \le i$;
(e) x_k is inserted between u_k and u_{k+1}, for all $i + 1 \le k < n$.

(6) Mode 6 derives v from u applying the next steps:

(a) $A_1, A_2, \dots, A_n$ are deleted;
(b) a central u_i is nondeterministically chosen, for some $0 \le i \le n$;
(c) x_j is inserted between u_j and u_{j+1}, for all $j < i$;
(d) x_k is inserted between u_{k-2} and u_{k-1}, for all $k > i + 1$.

(7) Mode 7 obtains v from u performing the steps stated below:

(a) $A_1, A_2, \dots, A_n$ are deleted;
(b) a central u_i is nondeterministically chosen, for some $0 \le i \le n$;

(c) x_j is inserted between u_{j-2} and u_{j-1}, for all $1 < j \leq i$;
(d) x_k is inserted between u_k and u_{k+1}, for all $i + 1 \leq k < n$.

(8) In mode 8, v is produced from u by following the given steps:

(a) $A_1, A_2, \ldots, A_n$ are deleted;
(b) x_1 and x_n are inserted into u_1 and u_{n-1}, respectively;
(c) x_i is inserted into $u_{i-1}u_i$, for all $1 < i < n$, to the right of x_{i-1} and to the left of x_{i+1}.

(9) Mode 9 derives v from u by the next procedure:

(a) $A_1, A_2, \ldots, A_n$ are deleted;
(b) x_1 and x_n are inserted into u_0 and u_n, respectively;
(c) x_i is inserted into $u_{i-1}u_i$, for all $1 < i < n$, to the right of x_{i-1} and to the left of x_{i+1}.

As obvious, all these jumping derivation modes reflect and formalize the above-described four-phase computation performed in a discontinuous way more adequately than their standard counterpart. Consequently, applications of these grammars are expected in any scientific area involving this kind of computation, ranging from applied mathematics through computational linguistics and compiler writing up to data mining and bioinformatics. In terms of the latter, a more specific application is described in the final section of this paper.

This paper is organized as follows. Section 2 gives all the necessary notation and terminology to follow the rest of the paper and formally introduces scattered context grammars. Then, Section 3 considers all the nine jumping modes, each of which is illustrated and investigated in a separate subsection. Most importantly, it is demonstrated that scattered context grammars working under any of the newly introduced derivation modes are computationally complete–that is, they characterize the family of recursively enumerable languages. Finally, Section 4 sketches application perspectives of scattered context grammars working under the nine jumping derivation modes and suggests four open problem areas to be discussed in the future.

2. Preliminaries and definitions

We assume that the reader is familiar with formal language theory (see [8, 9, 10, 11]). For a set W, $\mathrm{card}(W)$ denotes its cardinality. Let V be an alphabet—that is, a finite nonempty set. V^* denotes the set of all strings over V; algebraically, V^* represents the free monoid generated by V under the operation of concatenation. The identity element is denoted by ε. Set $V^+ = V^* - \{\varepsilon\}$; algebraically, V^+ is thus the free semigroup generated by V under the operation of concatenation. For $w \in V^*$, $|w|$ and $\mathrm{reversal}(w)$ denote the length of w and the reversal of w, respectively. For $L \subseteq V^*$, $\mathrm{reversal}(L) = \{\mathrm{reversal}(w) \mid w \in L\}$. The alphabet of w, denoted by $\mathrm{alph}(w)$, is the set of symbols appearing in w. For $v \in \Sigma$ and $w \in \Sigma^*$, $\mathrm{occur}(v, w)$ denotes the number of occurrences of v in w.

Let ϱ be a relation over V^*. The transitive and transitive—reflexive closure of ϱ are denoted by ϱ^+ and ϱ^*, respectively. Unless explicitly stated otherwise, we write $x \mathrel{\varrho} y$ instead $(x, y) \in \varrho$.

The family of recursively enumerable languages is denoted by $\mathbf{RE}$. Recall that scattered context grammars with erasing rules characterize $\mathbf{RE}$ (see [12]).

Definition 1. A *scattered context grammar* (an *SCG* for short) is a quadruple $G = (V, T, P, S)$, where V is an alphabet, $T \subset V$, set $N = V - T$, $S \in N$ is the *start symbol*, and $P \subseteq \bigcup_{m=1}^{\infty} N^m \times (V^*)^m$ is finite. V, T, and N are called the *total alphabet*, the *terminal alphabet*, and the *nonterminal alphabet*, respectively. P is called the set of *productions*. Instead of

$$(A_1, A_2, \ldots, A_n, x_1, x_2, \ldots, x_n) \in P,$$

where $A_i \in N$, $x_i \in V^*$, for $1 \leq i \leq n$, for some $n \geq 1$, we write

$$(A_1, A_2, \ldots, A_n) \rightarrow (x_1, x_2, \ldots, x_n).$$

Definition 2. Let $G = (V, T, P, S)$ be an SCG, and let ϱ be a relation over V^*. Set

$$\mathcal{L}(G, \varrho) = \{x \mid x \in T^*, S\ \varrho^*\ x\}.$$

$\mathcal{L}(G, \varrho)$ is said to be the *language that G generates by* ϱ. Set

$$SC(\varrho) = \{\mathcal{L}(G, \varrho) \mid G \text{ is an SCG}\}.$$

$SC(\varrho)$ is said to be *the language family that SCGs generate by* ϱ.

3. Results

This section is divided into nine subsections, each of which is dedicated to the discussion of one of the nine jumping derivation modes introduced in the previous section. More specifically, the section (1) repeats the definition of the mode in question, (2) illustrates it by an example, and (3) determines the generative power of SCGs using this mode. Most importantly, Section 4 demonstrates that scattered context grammars working under any of these newly introduced derivation modes are computationally complete–that is, they characterize the family of recursively enumerable languages.

Next, we give Lemma 3.1, which fulfills an important role in the proofs throughout Section 3. Its proof is to be found on page 307 in [13].

Lemma 3.1. Let $L \in \mathbf{RE}$. Then, there are alphabets Σ, Γ, a homomorphism $h : T^* \rightarrow \Sigma^*$, and two context-free languages L_1, L_2 such that $L = h(L_1 \cap L_2)$.

3.1. Jumping derivation mode 1

${}_1\!\Rightarrow$ represents, in fact, the ordinary scattered context derivation mode.

Definition 3. Let $G = (V, T, P, S)$ be an SCG. Let $u_0 A_1 u_1 \ldots A_n u_n \in V^*$ and $(A_1, A_2, \ldots, A_n) \rightarrow (x_1, x_2, \ldots, x_n) \in P$, for $n \geq 1$. Then,

$$u_0 A_1 u_1 A_2 u_2 \ldots A_n u_n \ {}_1\!\Rightarrow\ u_0 x_1 u_1 x_2 v_2 \ldots x_n u_n.$$

Example 3.2. Let $G = (V, T, P, S)$ be an SCG, where $V = \{S, S', S'', S''', A, B, C\ A', B', C', a, b, c\}$, $T = \{a, b, c\}$, and P contains the following rules:

(i) $(S) \to (aSA)$	(vii) $(S', C) \to (cS', C')$
(ii) $(S) \to (bSB)$	(viii) $(S', S'') \to (\varepsilon, S''')$
(iii) $(S) \to (cSC)$	(ix) $(S''', A') \to (S''', a)$
(iv) $(S) \to (S'S'')$	(x) $(S''', B') \to (S''', b)$
(v) $(S', A) \to (aS', A')$	(xi) $(S''', C') \to (S''', c)$
(vi) $(S', B) \to (bS', B')$	(xii) $(S''' \to \varepsilon)$

Consider ${}_1\!\Rightarrow$. Then, the derivation of G is as follows.

First, G generates any string $w \in T^*$ to the left of S and its reversal in capital letters to the right of S with linear productions. Then, it replaces S with $S'S''$. Next, while nondeterministically rewriting nonterminal symbols to the right of S'' to their prime versions, it generates the sequence of terminals in the same order to the left of S', which we denote w'. Since all the symbols to the right of S' must be rewritten, the sequence of symbols generated to the left of S' must have the same composition of symbols. Otherwise, no terminal string can be generated, so the derivation is blocked. Thereafter, S' is erased, and S'' is rewritten to S'''. Finally, the prime versions of symbols to the right of S''' are rewritten to the terminal string denoted w''. Consequently,

$$\mathcal{L}(G, {}_1\!\Rightarrow) = \big\{x \in T^* \mid x = ww'w'', w = \text{reversal}(w''), w' \text{ is any permutation of } w\big\}.$$

For instance, the string $abccabcba$ is generated by G in the following way:

$$\begin{aligned} S\ {}_1\!\Rightarrow aSA\ {}_1\!\Rightarrow abSBA\ {}_1\!\Rightarrow abcSCBA\ {}_1\!\Rightarrow abcS'S''CBA\ {}_1\!\Rightarrow abccS'S''C'BA \\ {}_1\!\Rightarrow abccaS'S''C'BA'\ {}_1\!\Rightarrow abccabS'S''C'B'A'\ {}_1\!\Rightarrow abccabS'''C'B'A' \\ {}_1\!\Rightarrow abccabS'''cB'A'\ {}_1\!\Rightarrow abccabS'''cbA'\ {}_1\!\Rightarrow abccabS'''cba\ {}_1\!\Rightarrow abccabcba \end{aligned}$$

Next, we prove that SCGs working under ${}_1\!\Rightarrow$ characterize $\mathbf{RE}$.

Theorem 3.3. $SC({}_1\!\Rightarrow) = \mathbf{RE}$.

The idea of the following proof is based on [14], however, we use significantly smaller grammar, since we do not study economical properties of jumping SCGs.

Proof:
As obvious, any SCG G can be turned to a Turing machine M so M accepts $\mathcal{L}(G, {}_1\!\Rightarrow)$. Thus, $SC({}_1\!\Rightarrow) \subseteq \mathbf{RE}$. Therefore, we only need to prove $\mathbf{RE} \subseteq SC({}_1\!\Rightarrow)$.

Let $L \in \mathbf{RE}$. Express $L = h(L_1 \cap L_2)$, where h, L_1, and L_2 have the same meaning as in Lemma 3.1. Since L_2 is context-free, so is $\text{reversal}(L_2)$ (see page 419 in [15]). Thus, there are context-free grammars G_1 and G_2 that generate L_1 and $\text{reversal}(L_2)$, respectively. More precisely, let $G_i = (V_i, T, P_i, S_i)$ for $i = 1, 2$. Let $T = \{a_1, \ldots, a_n\}$ and 0, 1, \$, $S \notin V_1 \cup V_2$ be the new symbols. Without any loss of generality, assume that $V_1 \cap V_2 = \emptyset$. Define the new morphisms

(I) $c : a_i \mapsto 10^i1$;

(II) $C_1 : V_1 \cup T \to V_1 \cup \Sigma \cup \{0,1\}^*$,
$$\begin{cases} A \mapsto A, & A \in V_1, \\ a \mapsto f(a), & a \in T; \end{cases}$$

(III) $C_2 : V_2 \cup T \to V_2 \cup \{0,1\}^*$,
$$\begin{cases} A \mapsto A, & A \in V_2, \\ a \mapsto c(a), & a \in T; \end{cases}$$

(IV) $f : a_i \mapsto h(a_i)c(a_i)$;

(V) $t : \Sigma \cup \{0,1,\$\} \to \Sigma$,
$$\begin{cases} a \mapsto a, & a \in \Sigma, \\ A \mapsto \varepsilon, & A \notin \Sigma; \end{cases}$$

(VI) $t' : \Sigma \cup \{0,1,\$\} \to \{0,1\}$,
$$\begin{cases} a \mapsto a, & a \in \{0,1\}, \\ A \mapsto \varepsilon, & A \notin \{0,1\}. \end{cases}$$

Finally, let $G = (V, \Sigma, P, S)$ be SCG, with $V = V_1 \cup V_2 \cup \{S, 0, 1, \$\}$ and P containing the rules

(1) $(S) \to (\$S_1 1111 S_2\$)$;
(2) $(A) \to (C_i(w))$, for all $A \to w \in P_i$, where $i = 1, 2$;
(3) $(\$, a, a, \$) \to (\varepsilon, \$, \$, \varepsilon)$, for $a = 0, 1$;
(4) $(\$) \to (\varepsilon)$.

Claim 1. $\mathcal{L}(G, {}_1\Rightarrow) = L$.

Proof:
Basic idea. First, the starting rule (1) is applied. The starting nonterminals S_1 and S_2 are inserted into the current sentential form. Then, by using the rules (2) G simulates derivations of both G_1 and G_2 and generates the sentential form $w = \$w_1 1111 w_2\$$.

Suppose $S \,{}_1\Rightarrow^* w$, where $\mathrm{alph}(w) \cap (N_1 \cup N_2) = \emptyset$. Recall, N_1 and N_2 denote the nonterminal alphabets of G_1 and G_2, respectively. If $t'(w_1) = \mathrm{reversal}(w_2)$, then $t(w_1) = h(v)$, where $v \in L_1 \cap L_2$ and $h(v) \in L$. In other words, w represents a successful derivation of both G_1 and G_2, where both grammars have generated the same sentence v; therefore G must generate the sentence $h(v)$.

The rules (3) serve to check, whether the simulated grammars have generated the identical words. Binary codings of the generated words are erased while checking the equality. Always the leftmost and the rightmost symbols are erased, otherwise some symbol is skipped. If the codings do not match, some 0 or 1 cannot be erased and no terminal string can be generated.

Finally, the symbols \$ are erased with the rule (4). If G_1 and G_2 generated the same sentence and both codings were successfully erased, G has generated the terminal sentence $h(v) \in L$. □

Claim 1 implies $\mathbf{RE} \subseteq SC({}_1\Rightarrow)$. Thus, Theorem 3.3 holds. □

3.2. Jumping derivation mode 2

Definition 4. Let $G = (V, T, P, S)$ be an SCG. Let $u = u_0 A_1 u_1 \ldots A_n u_n \in V^*$ and $(A_1, A_2, \ldots, A_n) \to (x_1, x_2, \ldots, x_n) \in P$, for $n \geq 1$. Then,

$$u_0 A_1 u_1 A_2 u_2 \ldots A_n u_n \;{}_2\Rightarrow\; v_0 x_1 v_1 x_2 v_2 \ldots x_n v_n,$$

where $u_0 u_1 \ldots u_n = v_0 v_1 \ldots v_n$, $u_0 z_1 = v_0$ and $z_2 u_n = v_n$, $z_1, z_2 \in V^*$.

Informally, by using $(A_1,A_2,\ldots,A_n) \to (x_1,x_2,\ldots,x_n) \in P$ G obtains $v = v_0x_1v_1x_2v_2\ldots x_nv_n$ from $u = u_0A_1u_1A_2u_2\ldots A_nu_n$ in ${}_2{\Rightarrow}$ as follows:

(1) $A_1, A_2, \ldots, A_n$ are deleted;
(2) x_1 through x_n are inserted in between u_0 and u_n.

Notice, the mutual order of inserted right-hand-side strings must be always preserved.

Example 3.4. Consider SCG defined in Example 3.2 and ${}_2{\Rightarrow}$. Context-free rules act in the same way as in ${}_1{\Rightarrow}$ unlike context-sensitive rules. Let us focus on the differences.

First, G generates the sentential form $wS'S''\overline{w}$, where $w \in T^*$ and $\overline{w}$ is the reversal of w in capital letters, with context-free derivations. Then, the nonterminals to the right of S' are rewritten to their prime versions and possibly randomly shifted closer to S', which may arbitrarily change their order. Additionally, the sequence of terminals in the same order is generated to the left of S', which we denote w'. S' may be also shifted, however, in such case it appears to the right of S'' and future application of the rule (viii) is excluded and no terminal string can be generated. Since all the symbols to the right of S' must be rewritten, the sequence generated to the left of S' must have the same composition of symbols. Next, S' is erased and S'' is rewritten to S''' at once, which ensures their mutual order is preserved. If any prime symbol occurs to the left of S''', it cannot be erased and the derivation is blocked. Finally, the prime versions of symbols to the right of S''' are rewritten to the terminal string denoted w'', which also enables random disordering. Consequently,

$$\mathcal{L}(G, {}_2{\Rightarrow}) = \{x \in T^* \mid x = ww'w'', w', w'' \text{ are any permutations of } w\}.$$

For example, the string $abcacbbac$ is generated by G in the following way:

$$\begin{aligned} S \;{}_2{\Rightarrow}\; aSA \;{}_2{\Rightarrow}\; abSBA \;{}_2{\Rightarrow}\; abcSCBA \;{}_2{\Rightarrow}\; abcS'S''CBA \;{}_2{\Rightarrow}\; abcaS'S''A'CB \\ {}_2{\Rightarrow}\; abcacS'S''A'C'B \;{}_2{\Rightarrow}\; abcacbS'S''B'A'C' \;{}_2{\Rightarrow}\; abcacbS'''B'A'C' \\ {}_2{\Rightarrow}\; abcacbS'''B'A'c \;{}_2{\Rightarrow}\; abcacbS'''bA'c \;{}_2{\Rightarrow}\; abcacbS'''bac \;{}_2{\Rightarrow}\; abcacbbac \end{aligned}$$

Theorem 3.5. $SC({}_2{\Rightarrow}) = \mathbf{RE}$.

Proof:
Clearly $SC({}_2{\Rightarrow}) \subseteq \mathbf{RE}$, so we only need to prove $\mathbf{RE} \subseteq SC({}_2{\Rightarrow})$.

Let $G = (V, \Sigma, P, S)$ be the SCG constructed in the proof of Theorem 3.3. First, we modify G to a new SCG G' so $\mathcal{L}(G, {}_1{\Rightarrow}) = \mathcal{L}(G', {}_1{\Rightarrow})$. Then, we prove $\mathcal{L}(G', {}_2{\Rightarrow}) = \mathcal{L}(G', {}_1{\Rightarrow})$.

Construction. Set

$$N = \{\lceil, \rceil, \lfloor, \rfloor, |, X, \underline{X}, \overline{X}, \underline{\overline{X}}, Y, \underline{Y}, \overline{Y}, \underline{\overline{Y}}\}$$

where $V \cap N = \emptyset$. Define the new morphisms

(I) $\overline{C}_1 : V_1 \cup T,$
$$\begin{cases} A \mapsto A, & A \in V_1, \\ a \mapsto \lceil f(a) \rceil\ |, & a \in T; \end{cases}$$

(II) $\overline{C}_2 : V_2 \cup T,$
$$\begin{cases} A \mapsto A & A \in V_2, \\ a \mapsto |\ \lceil c(a) \rceil, & a \in T; \end{cases}$$

(III) $b : \Sigma \cup \{0,1,\$\} \cup N \to \{0,1\},$
$$\begin{cases} A \mapsto A, & A \in \{0,1\}, \\ A \mapsto \varepsilon, & A \notin \{0,1\}. \end{cases}$$

(IV) $\overline{t'} : \Sigma \cup \{0,1,\$\} \cup N \to \{0,1,\$\} \cup N,$
$$\begin{cases} A \mapsto A, & A \in \{\$\} \cup N, \\ A \mapsto t'(A), & A \notin \{\$\} \cup N. \end{cases}$$

Let $G' = (V', \Sigma, P', S)$ be SCG, with $V' = V \cup N$ and P' containing

(1) $(S) \to (\rceil X\$S_1\lceil 11 \,||\, 11\rceil S_2\$Y\lceil);$
(2) $(A) \to (\overline{C}_i(w))$ for $A \to w \in P_i$, where $i = 1, 2;$
(3) $(\rceil, X, \lceil) \to (\rfloor, \underline{\overline{X}}, \rfloor)$,$(\rceil, Y, \lceil) \to (\lfloor, \underline{\overline{Y}}, \lfloor);$
(4) $(\rfloor, \underline{\overline{X}}, \rfloor) \to (\rfloor, \overline{X}, \rfloor)$,$(\lfloor, \underline{\overline{Y}}, \lfloor) \to (\lfloor, \overline{Y}, \lfloor);$
(5) $(\$, 0, \overline{X}, \overline{Y}, 0, \$) \to (\varepsilon, \$, \underline{\overline{X}}, \underline{\overline{Y}}, \$, \varepsilon);$
(6) $(\$, \overline{X}, \overline{Y}, \$) \to (\varepsilon, \overline{X}\$, \$\overline{Y}, \varepsilon);$
(7) $(\rfloor, \overline{X}, \$, \rfloor, \lfloor, \$, \overline{Y}, \lfloor) \to (\varepsilon, \varepsilon, \varepsilon, \underline{X}\$, \$\underline{Y}, \varepsilon, \varepsilon, \varepsilon);$
(8) $(\underline{X}, 1, 1, |, |, 1, 1, \underline{Y}) \to (\varepsilon, \varepsilon, \varepsilon, X, Y, \varepsilon, \varepsilon, \varepsilon);$
(9) $(\$) \to (\varepsilon)$, $(X) \to (\varepsilon)$, $(Y) \to (\varepsilon)$.

Notice that X and Y hold the current state of computation and force the context-sensitive rules to be used in the following order:

(a) after applying the rule 3, only the rule 4 may be applied;
(b) after applying the rule 4, only the rule 5 or 6 may be applied;
(c) after applying the rule 5, only the rule 4 may be applied;
(d) after applying the rule 6, only the rule 7 may be applied;
(e) after applying the rule 7, only the rule 8 may be applied;
(f) after applying the rule 8, only the rule 3 may be applied.

Claim 2. $\mathcal{L}(G', {}_1\Rightarrow) = \mathcal{L}(G, {}_1\Rightarrow)$.

Proof:
The context-free rules (1) and (2) of G' correspond one to one to the rules (1) and (2) of G, only the codings of terminals contain additional symbols. Thus, for every derivation in G

$$S \ {}_1\Rightarrow^* \ \$v_1 1111 v_2\$ = v,$$

where v is generated by using the rules (1) and (2) and $\mathrm{alph}(v) \cap (N_1 \cup N_2) = \emptyset$, there is

$$S \ {}_1\Rightarrow^* \rceil X\$w_1\lceil 11 \,||\, 11\rceil w_2\$Y\lceil = w$$

in G' generated by the rules (1) and (2), where $b(w_1) = t'(v_1)$, $b(w_2) = v_2$. This also holds vice versa. Since such a sentential form represents a successful derivations of both G_1 and G_2, without any loss of generality, we can consider it in every successful derivation of either G, or G'. Additionally, in G

$$v\ {}_1\!\Rightarrow^*\ v', v' \in \Sigma^*$$

if and only if $t'(v_1) = \text{reversal}(v_2)$. Note, $v' = t(v)$. Therefore, we have to prove

$$w\ {}_1\!\Rightarrow^*\ w', w' \in \Sigma^*$$

if and only if $\overline{t'}(w_1) = \text{reversal}(w_2)$. Then $v' = w'$.

Claim 3. In G', for

$$S\ {}_1\!\Rightarrow^*\ \rceil X\$ w_1 \lceil 11 \mid\mid 11 \rceil w_2 \$ Y \lceil = w, \text{alph}(w) \cap (N_1 \cup N_2) = \emptyset,$$

where w is generated by using the rules (1) and (2),

$$w\ {}_1\!\Rightarrow^*\ w',$$

where $w' \in \Sigma^*$ if and only if $\overline{t'}(w_1) = \text{reversal}(w_2)$.

For the sake of readability, in the next proof we omit all symbols from Σ in w_1—that is, we consider only nonterminal symbols, which are to be erased.

Proof:
If. Suppose $w_1 = \text{reversal}(w_2)$, then $w\ {}_1\!\Rightarrow^*\ \varepsilon$. From the construction of G',

$$w_1 = (\lceil 10^{i_1} 1 \rceil \mid)(\lceil 10^{i_2} 1 \rceil \mid) \dots (\lceil 10^{i_n} 1 \rceil \mid),$$

where $i_j \in \{1, \dots, |\Sigma|\}$, $1 \leq j \leq n$, $n \geq 0$. Consider two cases—(I) $n = 0$ and (II) $n \geq 1$.

(I) If $n = 0$, $w = \rceil X\$\lceil 11 \mid\mid 11 \rceil \$ Y \lceil$. Then, by using the rules (3) and (4), the rules (7) and (8), and four times the rules (9), we obtain

$$\begin{array}{l}
\rceil X\$\lceil 11 \mid\mid 11 \rceil \$ Y \lceil\ {}_1\!\Rightarrow \rfloor \underline{\overline{X}}\$\rfloor 11 \mid\mid 11 \rceil \$ Y \lceil\ {}_1\!\Rightarrow \\
\rfloor \underline{\overline{X}}\$\rfloor 11 \mid\mid 11 \lfloor \$ \underline{\overline{Y}} \lfloor\ {}_1\!\Rightarrow \rfloor \overline{X}\$\rfloor 11 \mid\mid 11 \lfloor \$ \underline{\overline{Y}} \lfloor\ {}_1\!\Rightarrow \\
\rfloor \overline{X}\$\rfloor 11 \mid\mid 11 \lfloor \$ \overline{Y} \lfloor\ {}_1\!\Rightarrow\ \underline{X}\$ 11 \mid\mid 11 \$ \underline{Y} \qquad {}_1\!\Rightarrow \\
\$ XY \$\ {}_1\!\Rightarrow\ XY\$\ {}_1\!\Rightarrow\ Y\$\ {}_1\!\Rightarrow\ \$\ {}_1\!\Rightarrow\ \varepsilon
\end{array}$$

and the claim holds.

(II) Let $n \geq 1$,

$$\begin{aligned}
w &= \rceil X\$\lceil 10^{i'} 1 \rceil \mid (\lceil 10^{i_m} 1 \rceil \mid)^k \lceil 11 \mid\mid 11 \rceil (\mid \lceil 10^{j_{m'}} 1 \rceil)^k \mid \lceil 10^{j'} 1 \rceil \$ Y \lceil \\
&= \rceil X\$\lceil 10^{i'} 1 \rceil \mid u \mid \lceil 10^{j'} 1 \rceil \$ Y \lceil
\end{aligned}$$

where $k \geq 0$, $m, m' \in \{1, \dots, k\}$, $i', i_m, j', j_{m'} \in \{1, \dots, |\Sigma|\}$. Sequentially using both rules (3) and (4) and the rule (7) we obtain the derivation

$$\begin{array}{l}\rceil X\$\lceil 10^{i'}1\rceil \mid u \mid \lceil 10^{j'}1\rceil\$Y\lceil\ {}_1\!\Rightarrow \rfloor\underline{\overline{X}}\$\rfloor 10^{i'}1\rceil \mid u \mid \lceil 10^{j'}1\rceil\$Y\lceil\ {}_1\!\Rightarrow \\ \rfloor\underline{\overline{X}}\$\rfloor 10^{i'}1\rceil \mid u \mid \lceil 10^{j'}1\lfloor\$\underline{\overline{Y}}\lfloor\ {}_1\!\Rightarrow \rfloor\overline{X}\$\rfloor 10^{i'}1\rceil \mid u \mid \lceil 10^{j'}1\lfloor\$\underline{\overline{Y}}\lfloor\ {}_1\!\Rightarrow \\ \rfloor\overline{X}\$\rfloor 10^{i'}1\rceil \mid u \mid \lceil 10^{j'}1\lfloor\$\overline{Y}\lfloor\ {}_1\!\Rightarrow \underline{X}\$10^{i'}1\rceil \mid u \mid \lceil 10^{j'}1\$\underline{Y}\end{array}$$

Next, we prove

$$w' = \underline{X}\$10^{i'}1\rceil \mid (\lceil 10^{i_m}1\rceil \mid)^k\lceil 11 \mid\mid 11\rceil(\mid \lceil 10^{j_{m'}}1\rceil)^k \mid \lceil 10^{j'}1\$\underline{Y}\ {}_1\!\Rightarrow^* \varepsilon$$

by induction on $k \geq 0$.

Basis. Let $k = 0$. Then,

$$w' = \underline{X}\$10^{i'}1\rceil \mid \lceil 11 \mid\mid 11\rceil \mid \lceil 10^{j'}1\$\underline{Y}.$$

By using a rule (8) and twice a rule (3) G' performs

$$\begin{array}{ll}\underline{X}\$10^{i'}1\rceil \mid \lceil 11 \mid\mid 11\rceil \mid \lceil 10^{j'}1\$\underline{Y}\ {}_1\!\Rightarrow \$0^{i'}\rceil X\lceil 11 \mid\mid 11\rceil Y\lceil 0^{j'}\$ & \\ {}_1\!\Rightarrow \$0^{i'}\rfloor\underline{\overline{X}}\rfloor 11 \mid\mid 11\rceil Y\lceil 0^{j'}\$ & {}_1\!\Rightarrow \$0^{i'}\rfloor\underline{\overline{X}}\rfloor 11 \mid\mid 11\lfloor\underline{\overline{Y}}\lfloor 0^{j'}\$\end{array}$$

Since $i' = j'$, both sequences of 0s are simultaneously erased by repeatedly using both rules (4) and the rule (5). Observe that

$$\$0^{i'}\rfloor\underline{\overline{X}}\rfloor 11 \mid\mid 11\lfloor\underline{\overline{Y}}\lfloor 0^{j'}\$\ {}_1\!\Rightarrow^* \$\rfloor\overline{X}\rfloor 11 \mid\mid 11\lfloor\overline{Y}\lfloor\$$$

Finally, by applying the rules (4), (6), (7), (8), and (9), we finish the derivation as

$$\begin{array}{c}\$\rfloor\overline{X}\rfloor 11 \mid\mid 11\lfloor\overline{Y}\lfloor\$\ {}_1\!\Rightarrow \rfloor\overline{X}\$\rfloor 11 \mid\mid 11\lfloor\$\overline{Y}\lfloor\ {}_1\!\Rightarrow \\ \underline{X}\$11 \mid\mid 11\$\underline{Y}\ {}_1\!\Rightarrow \$XY\$\ {}_1\!\Rightarrow^* \varepsilon\end{array}$$

and the basis holds.

Induction Hypothesis. Suppose there exists $k \geq 0$ such that

$$w' = \underline{X}\$10^{i'}1\rceil \mid (\lceil 10^{i_m}1\rceil \mid)^l\lceil 11 \mid\mid 11\rceil(\mid \lceil 10^{j_{m'}}1\rceil)^l \mid \lceil 10^{j'}1\$\underline{Y}\ {}_1\!\Rightarrow^* \varepsilon,$$

where $m, m' \in \{1, \dots, l\}$, $i', i_m, j', j_{m'} \in \{1, \dots, |\Sigma|\}$, for all $0 \leq l \leq k$.

Induction Step. Consider any

$$w' = \underline{X}\$10^{i'}1\rceil \mid (\lceil 10^{i_m}1\rceil \mid)^{k+1}\lceil 11 \mid\mid 11\rceil(\mid \lceil 10^{j_{m'}}1\rceil)^{k+1} \mid \lceil 10^{j'}1\$\underline{Y},$$

where $m, m' \in \{1, \dots, k+1\}$, $i', i_m, j', j_{m'} \in \{1, \dots, |\Sigma|\}$. Since $k + 1 \geq 1$

$$\begin{array}{rl}w' &= \underline{X}\$10^{i'}1\rceil \mid \lceil 10^{i''}1\rceil \mid u \mid \lceil 10^{j''}1\rceil \mid \lceil 10^{j'}1\$\underline{Y} \\ u &= (\lceil 10^{i_m}1\rceil \mid)^k\lceil 11 \mid\mid 11\rceil(\mid \lceil 10^{j_{m'}}1\rceil)^k\end{array}$$

By using the rule (8) and both rules (3) G' performs

$$\begin{array}{ll} \underline{X}\$10^{i'}1\rceil \mid \lceil 10^{i''}1\rceil \mid u \mid \lceil 10^{j''}1\rceil \mid \lceil 10^{j'}1\$\underline{Y} & {}_1\Rightarrow \\ \$0^{i'}\rceil X\lceil 10^{i''}1\rceil \mid u \mid \lceil 10^{j''}1\rceil Y\lceil 0^{j'}\$ & {}_1\Rightarrow \\ \$0^{i'}\rfloor \overline{X}\rfloor 10^{i''}1\rceil \mid u \mid \lceil 10^{j''}1\rceil Y\lceil 0^{j'}\$ & {}_1\Rightarrow \\ \$0^{i'}\rfloor \overline{X}\rfloor 10^{i''}1\rceil \mid u \mid \lceil 10^{j''}1\lfloor \overline{Y}\lfloor 0^{j'}\$ & \end{array}$$

Since $i' = j'$, the prefix of 0s and the suffix of 0s are simultaneously erased by repeatedly using the rules (4) and the rule (5).

$$\$0^{i'}\rfloor \overline{X}\rfloor 10^{i''}1\rceil \mid u \mid \lceil 10^{j''}1\lfloor \overline{Y}\lfloor 0^{j'}\$ \ {}_1\Rightarrow^* \ \$\rfloor \overline{X}\rfloor 10^{i''}1\rceil \mid u \mid \lceil 10^{j''}1\lfloor \overline{Y}\lfloor \$$$

Finally, G' uses the rule (6) and the rule (7)

$$\begin{array}{c} \$\rfloor \overline{X}\rfloor 10^{i''}1\rceil \mid u \mid \lceil 10^{j''}1\lfloor \overline{Y}\lfloor \$ \ {}_1\Rightarrow \ \rfloor \overline{X}\$\rfloor 10^{i''}1\rceil \mid u \mid \lceil 10^{j''}1\lfloor \$\overline{Y}\lfloor \ {}_1\Rightarrow \\ \underline{X}\$10^{i''}1\rceil \mid u \mid \lceil 10^{j''}1\$\underline{Y} = w'' \end{array}$$

where

$$w'' = \underline{X}\$10^{i''}1\rceil \mid (\lceil 10^{i_m}1\rceil \mid)^k \lceil 11 \mid\mid 11\rceil (\mid \lceil 10^{j_{m'}}1\rceil)^k \mid \lceil 10^{j''}1\$\underline{Y}.$$

By induction hypothesis, $w''{}_1\Rightarrow^* \varepsilon$, which completes the proof.

Only if. Suppose that $w_1 \neq \text{reversal}(w_2)$, then there is no w' satisfying $w\ {}_1\Rightarrow^* w'$ and $w' = \varepsilon$.

From the construction of G', there is no rule shifting the left \$ to the left and no rule shifting the right \$ to the right. Since the rule (5) is the only one erasing 0s and these 0s must occur between two \$s, if there is any 0, which is not between the two \$s, it is unable to be erased. Moreover, an application of the rule (5) moves the left \$ on the previous position of erased left 0; if it is not the leftmost, the derivation is blocked. It is symmetric on the right. A similar situation is regarding 1s, X, and Y. Thus, for the sentential form w, if 0 or 1 is the rightmost or the leftmost symbol of w, no terminal string can be generated.

Since $w_1 \neq \text{reversal}(w_2)$, the codings of terminal strings generated by G_1 and G_2 are different. Then, there is a and a', where $w_1 = vau$, $w_2 = u'a'v$, and $a \neq a'$. For always the outermost 0 or 1 is erased, otherwise the derivation is blocked, suppose the derivation correctly erases both strings v, so a and a' are the outermost symbols. The derivation can continue in the following two ways.

(I) Suppose the outermost 0s are erased before the outermost 1s. Then, the rule (5) is used, which requires $\overline{X}$ and $\overline{Y}$ between the previous positions of 0s. However, there is 1, a or a', which is not between X and Y.

(II) Suppose the outermost 1s are erased before the outermost 0s. Then, the rule (8) is used, which requires $\underline{X}$ and $\underline{Y}$ in the current sentential form. The symbols $\underline{X}$ and $\underline{Y}$ are produced by the rule (7), which requires X and \$ between two symbols $\rfloor$ and Y and \$ between two symbols $\lfloor$. Suppose w' is the current sentential form. Since w_1 or $\text{reversal}(w_2)$ is of the form

$$\ldots \lceil 10^{i_0}1\rceil \mid \lceil 10^{i_1}1\rceil \mid \lceil 10^{i_2}1\rceil \mid \ldots,$$

where $i_0, i_1, i_2 \in \{1, \dots, |\Sigma|\}$, there is 0 as the leftmost or rightmost symbol of w' and $X\$$ and $\$Y$ occurs between $\rfloor$s and $\lfloor$s, respectively. However, this 0 is obviously not between the two \$s and remains permanently in the sentential form.

We showed that G' can generate the terminal string from the sentential form w if and only if $\overline{t'}(w_1) = \text{reversal}(w_2)$, so the claim holds. □

We proved that for any $w \in \Sigma^*$, $S \;_1\!\Rightarrow^* \; w$ in G if and only if $S \;_1\!\Rightarrow^* \; w$ in G', and Claim 2 holds. □

Let us turn to ${}_2\!\Rightarrow$.

Claim 4. $\mathcal{L}(G', {}_2\!\Rightarrow) = \mathcal{L}(G', {}_1\!\Rightarrow)$.

Proof:
In ${}_2\!\Rightarrow$, applications of context-free rules progress in the same way as in ${}_1\!\Rightarrow$. While using context-sensitive rules inserted right-hand-side strings can be nondeterministically scattered between the previous positions of the leftmost and rightmost affected nonterminals, only their order is preserved. We show, we can control this by the construction of G'.

Recall the observations made at the beginning of the proof of Claim 2. Since the behaviour of context-free rules remains unchanged in terms of ${}_2\!\Rightarrow$, these still hold true. It remains to prove that Claim 3 also holds in ${}_2\!\Rightarrow$.

In a special case, ${}_2\!\Rightarrow$ can behave exactly as ${}_1\!\Rightarrow$, hence definitely $\mathcal{L}(G', {}_1\!\Rightarrow) \subseteq \mathcal{L}(G', {}_2\!\Rightarrow)$. We prove

$$w \notin \mathcal{L}(G', {}_1\!\Rightarrow) \Rightarrow w \notin \mathcal{L}(G', {}_2\!\Rightarrow).$$

Therefore, to complete the proof of Claim 4, we establish the following claim.

Claim 5. In G', for

$$S \;_1\!\Rightarrow^* \rceil X\$w_1\lceil 11 \mid\mid 11\rceil w_2\$Y\lceil = w, \text{alph}(w) \cap (N_1 \cup N_2) = \emptyset,$$

where w is generated only by using the rules (1) and (2), and $\overline{t'}(w_1) \neq \text{reversal}(w_2)$, there is no w', where

$$w \;_1\!\Rightarrow^* \; w', w' \in \Sigma^*.$$

For the sake of readability, in the next proof we omit all symbols from Σ in w_1—we consider only nonterminal symbols, which are to be erased.

Proof:
Suppose any w, where

$$S \;_1\!\Rightarrow^* \; w = \rceil X\$w_1\lceil 11 \mid\mid 11\rceil w_2\$Y\lceil$$

in G' and w is generated by using the rules (1) and (2), $\text{alph}(w) \cap (N_1 \cup N_2) = \emptyset$, and $w_1 \neq \text{reversal}(w_2)$.

From the construction of G', there is no rule shifting the left \$ to the left and no rule shifting the right \$ to the right. Neither ${}_2\Rightarrow$ can do this. Since the rule (5) is the only one erasing 0s and these 0s must be between two \$s, if there is any 0, which is not between the two \$s, it cannot be erased. A similar situation is regarding 1s, X, and Y. Thus, for the sentential form w, if 0 or 1 is the outermost symbol of w, no terminal string can be generated.

Consider two cases (I) $w_1 = \varepsilon$ or $w_2 = \varepsilon$ and (II) $w_1 \neq \varepsilon$ and $w_2 \neq \varepsilon$.

(I) Suppose the condition does not apply. Without any loss of generality, suppose $w_1 = \varepsilon$. Since $w_1 \neq \text{reversal}(w_2)$, $w_2 \neq \varepsilon$. Then,

$$w = \rceil X\$\lceil 11 \mid\mid 11\rceil(\mid \lceil 10^{i_m}1\rceil)^k \mid \lceil 10^{i'}1\rceil\$Y\lceil,$$

where $k \geq 0$, $m \in \{1, \dots, k\}$, $i_m, i' \in \{1, \dots, |\Sigma|\}$.

First, the rules (3) and (9) are the only applicable, however, application of the rule (9) would block the derivation, so we do not consider it. While rewriting X, the leftmost $\rceil$ is rewritten. Unless the leftmost $\lceil$ is chosen, it becomes unpaired and, thus, cannot be erased. It is symmetric with Y. After the application of the rules (3), the rules (4) becomes applicable. The positions of the symbols \$ must be preserved for future usage of the rule (7). Then, the only way of continuing a successful derivation is

$$\begin{array}{l}
\rceil X\$\lceil 11 \mid\mid 11\rceil(\mid \lceil 10^{i_m}1\rceil)^k \mid \lceil 10^{i'}1\rceil\$Y\lceil \;{}_2\Rightarrow \\
\rfloor\underline{\overline{X}}\$\rfloor 11 \mid\mid 11\rceil(\mid \lceil 10^{i_m}1\rceil)^k \mid \lceil 10^{i'}1\rceil\$Y\lceil \;{}_2\Rightarrow \\
\rfloor\underline{\overline{X}}\$\rfloor 11 \mid\mid 11\rceil(\mid \lceil 10^{i_m}1\rceil)^k \mid \lceil 10^{i'}1\lfloor\$\underline{\overline{Y}}\lfloor \;{}_2\Rightarrow \\
\rfloor\overline{X}\$\rfloor 11 \mid\mid 11\rceil(\mid \lceil 10^{i_m}1\rceil)^k \mid \lceil 10^{i'}1\lfloor\$\underline{\overline{Y}}\lfloor \;{}_2\Rightarrow \\
\rfloor\overline{X}\$\rfloor 11 \mid\mid 11\rceil(\mid \lceil 10^{i_m}1\rceil)^k \mid \lceil 10^{i'}1\lfloor\$\overline{Y}\lfloor
\end{array}$$

Notice that if neighboring nonterminals are rewritten, ${}_2\Rightarrow$ do not shift any symbol.

Next, the rule (7) is the only applicable possibly shifting $\underline{X}$, $\underline{Y}$, and \$s anywhere into the current sentential form. However, if any shift is performed, there is a symbol 1 as the outer most symbol, which is obviously unable to be erased. Thus,

$$\rfloor\overline{X}\$\rfloor 11 \mid\mid 11\rceil(\mid \lceil 10^{i_m}1\rceil)^k \mid \lceil 10^{i'}1\lfloor\$\overline{Y}\lfloor {}_2\Rightarrow \underline{X}\$11 \mid\mid 11\rceil(\mid \lceil 10^{i_m}1\rceil)^k \mid \lceil 10^{i'}1\$\underline{Y} = w'$$

Next, consider two cases depending on k.

(i) Suppose $k = 0$. Then,

$$w' = \underline{X}\$11 \mid\mid 11\rceil \mid \lceil 10^{i'}1\$\underline{Y}.$$

Since $i' > 0$, the rule (5) must be used. It requires presence of $\overline{X}$ and $\overline{Y}$ in the current sentential form. These can be obtained only by application of the rule (8) and both rules from (3) and (4). However, it must rewrite two pairs of $\rceil, \lceil$, but there is only one remaining. Therefore, there are i' symbols 0, which cannot be erased, and no terminal string can be generated.

(ii) Suppose $k > 0$. Then, w' is of the form

$$\underline{X}\$11 \mid\mid 11\rceil \mid \lceil u\rceil \mid \lceil 10^{i'}1\$\underline{Y}.$$

The rule (8) is the only applicable. It rewrites $\underline{X}$ to X, $\underline{Y}$ to Y and put them potentially anywhere into the current sentential form. However, the rules (3), which are the only containing X and Y on the left-hand side, require X and Y situated between $\rceil$ and $\lceil$.

$$\underline{X}\$11 \mid\mid 11\rceil \mid \lceil u\rceil \mid \lceil 10^{i'}1\$\underline{Y} \;{}_2\!\Rightarrow\; \$11 \mid\mid 11\rceil X\lceil u\rceil Y\lceil 0^{i'}\$$$

Without any loss of generality, we omit other possibilities of erasing the symbols $\mid$ or 1, because the derivation would be blocked in the same way. Since there is no 0 to the left of X, the future application of the rule (5) is excluded and the rightmost sequence of 0s is obviously skipped and cannot be erased any more.

(II) Suppose the condition applies. Then,

$$\begin{aligned} w &= \rceil X\$\lceil 10^{i}1\rceil \mid (\lceil 10^{j_m}1\rceil \mid)^{k}\lceil 11 \mid\mid 11\rceil(\mid \lceil 10^{j_{m'}}1\rceil)^{k'} \mid \lceil 10^{i'}1\rceil\$Y\lceil \\ &= \rceil X\$\lceil 10^{i}1\rceil \mid \lceil u\rceil \mid \lceil 10^{i'}1\rceil\$Y\lceil \end{aligned}$$

where $k, k' \geq 0$, $m \in \{1, \ldots, k\}$, $m' \in \{1, \ldots, k'\}$, $i_m, i'_m, j, j' \in \{1, \ldots, |\Sigma|\}$.

First, the situation is completely the same as in (I), the only possibly non-blocking derivation consists of application of both rules (3) and (4) followed by application of the rule (7). No left-hand-side string may be shifted during the application of these rules or the derivation is blocked.

$$\begin{aligned} &\rceil X\$\lceil 10^{i}1\rceil \mid \lceil u\rceil \mid \lceil 10^{i'}1\rceil\$Y\lceil \;{}_2\!\Rightarrow\; \rfloor\underline{\overline{X}}\$\rfloor 10^{i}1\rceil \mid \lceil u\rceil \mid \lceil 10^{i'}1\rceil\$Y\lceil \;{}_2\!\Rightarrow \\ &\rfloor\underline{\overline{X}}\$\rfloor 10^{i}1\rceil \mid \lceil u\rceil \mid \lceil 10^{i'}1\lfloor\$\underline{\overline{Y}}\lfloor \;{}_2\!\Rightarrow\; \rfloor\overline{X}\$\rfloor 10^{i}1\rceil \mid \lceil u\rceil \mid \lceil 10^{i'}1\lfloor\$\underline{\overline{Y}}\lfloor \;{}_2\!\Rightarrow \\ &\rfloor\overline{X}\$\rfloor 10^{i}1\rceil \mid \lceil u\rceil \mid \lceil 10^{i'}1\lfloor\$\overline{Y}\lfloor \;{}_2\!\Rightarrow\; \underline{X}\$10^{i}1\rceil \mid \lceil u\rceil \mid \lceil 10^{i'}1\$\underline{Y} \end{aligned}$$

Next, the rule (8) is the only applicable rule, which erases four symbols 1, two $\mid$, rewrites $\underline{X}$ to X and $\underline{Y}$ to Y, and inserts them possibly anywhere into the current sentential form. However, X must be inserted between $\rceil$ and $\lceil$, otherwise the rule (3) is not applicable and X remains permanently in the sentential form. Unless the leftmost pair of $\rceil$ and $\lceil$ is chosen, there are skipped symbols 1 remaining to the left of X. The rules (6) and (7) ensures the derivation is blocked, if X is shifted to the right. Additionally, the only way to erase 1s is the rule (8), but these 1s must be to the right of X. Thus, the skipped symbols 1 cannot be erased. Therefore, the pair of $\rceil$ and $\lceil$ is the leftmost or the derivation is blocked. Moreover, the two erased 1s are also the leftmost or they cannot be erased in the future and the same holds for the left erased symbol $\mid$. A similar situation is regarding Y. Then,

$$\underline{X}\$10^{i}1\rceil \mid \lceil u\rceil \mid \lceil 10^{i'}1\$\underline{Y} \;{}_2\!\Rightarrow\; \$0^{i}\rceil X\lceil u\rceil Y\lceil 0^{i'}\$$$

and by using the rules (3) and repeatedly the rules (4) and (5) both outer most sequences of 0s can be erased, if $i = i'$. Additionally, the rules (4) ensure, X and Y are never shifted. If there is any 0 skipped, it cannot be erased and the derivation is blocked.

$$\$0^i\rceil X\lceil u\rceil Y\lceil 0^{i'}\$ \;{}_2\Rightarrow^* \; \$0^i\rfloor\overline{\underline{X}}\rfloor u\lfloor\overline{\underline{Y}}\lfloor 0^{i'}\$ \;{}_2\Rightarrow^* \; \$\rfloor\overline{X}\rfloor u\lfloor\overline{Y}\lfloor\$$$

Finally, by the rules (6) and (7) both terminal codings can be completely erased and $\underline{X}$, $\underline{Y}$, and two \$ are the outermost symbols, if no symbol is skipped.

$$\$\rfloor\overline{X}\rfloor u\lfloor\overline{Y}\lfloor\$ \;{}_2\Rightarrow\; \rfloor\underline{X}\$\rfloor u\lfloor\$\underline{Y}\lfloor \;{}_2\Rightarrow\; \underline{X}\$u\$\underline{Y}$$

Since $w_1 \neq w_2$, $w_1 = vau$ and $w_2 = u'a'v$, where $a \neq a'$ are the outermost non-identical terminal codings. Derivation can always erase vs, as it was described, or be blocked before. Without any loss of generality, we have to consider two cases.

(i) Suppose $au = \varepsilon$. Then, $u'a' \neq \varepsilon$ and the situation is the same as in (I), no terminal string can be generated and the derivation is blocked.

(ii) Suppose $au \neq \varepsilon$, $u'a' \neq \varepsilon$. If the derivation is not blocked before, it may generate the sentential form

$$\$0^i\rceil X\lceil u\rceil Y\lceil 0^{i'}\$,$$

where $10^i1 = a$, $10^{i'}1 = a'$. Then, $i \neq i'$ and while simultaneously erasing the sequences of 0s of both codings, one is erased before the second one. The rule (5) becomes inapplicable and there is no way not to skip the remaining part of the second sequence of 0s. The derivation is blocked.

We covered all possibilities and showed, there is no way to generate terminal string $w' \notin \mathcal{L}(G', {}_1\Rightarrow)$, and the claim holds. □

Since $S \;{}_1\Rightarrow^* w$, $w \in \Sigma^*$ if and only if $S \;{}_2\Rightarrow^* w$, Claim 4 holds. □

We proved $\mathcal{L}(G', {}_2\Rightarrow) = \mathcal{L}(G', {}_1\Rightarrow)$, $\mathcal{L}(G', {}_1\Rightarrow) = \mathcal{L}(G, {}_1\Rightarrow)$, and $\mathcal{L}(G, {}_1\Rightarrow) = L$, then $\mathcal{L}(G', {}_2\Rightarrow) = L$, so the proof of Theorem 3.5 is completed. □

3.3. Jumping derivation mode 3

Definition 5. Let $G = (V, T, P, S)$ be an SCG. Let $u = u_0A_1u_1 \dots A_nu_n \in V^*$ and $(A_1,A_2, \dots, A_n) \to (x_1,x_2,\dots,x_n) \in P$, for $n \geq 1$. Then,

$$u_0A_1u_1A_2u_2 \dots A_nu_n \;{}_3\Rightarrow\; v_0x_1v_1x_2v_2 \dots x_nv_n,$$

where $u_0u_1 \dots u_n = v_0v_1 \dots v_n$, $u_0 = v_0z_1$ and $u_n = z_2v_n$, $z_1, z_2 \in V^*$.

Informally, G obtains $v = v_0x_1v_1x_2v_2 \dots x_nv_n$ from $u = u_0A_1u_1A_2u_2 \dots A_nu_n$ by $(A_1,A_2, \dots, A_n) \to (x_1,x_2,\dots,x_n) \in P$ in terms of ${}_3\Rightarrow$ as follows:

(1) $A_1, A_2, \dots, A_n$ are deleted;
(2) x_1 and x_n are inserted into u_0 and u_n, respectively;
(3) x_2 through x_{n-1} are inserted in between the newly inserted x_1 and x_n.

Example 3.6. Let $G = (V, T, P, S)$, where $V = \{S, A, \$, a, b\}$, $T = \{a, b\}$, be an SCG with P containing the following rules:

$$\begin{array}{rl} \text{(i)} & (S) \rightarrow (A\$) \\ \text{(ii)} & (A) \rightarrow (aAb) \\ \text{(iii)} & (A, \$) \rightarrow (A, \$) \end{array} \qquad \begin{array}{rl} \text{(iv)} & (A) \rightarrow (\varepsilon) \\ \text{(v)} & (\$) \rightarrow (\varepsilon) \end{array}$$

Context-free rules are not influenced by ${}_3\!\Rightarrow$. Therefore, after applying starting rule (i), G generates $a^n b^n$, where $n \geq 0$, by using the rule (ii) or finishes the derivation with rules (iv) and (v). However, at any time during the derivation the rule (iii) can be applied. It inserts or erases nothing, but with ${}_3\!\Rightarrow$ it potentially shifts A to the left. Notice, the symbol \$ is always the rightmost and, thus, cannot be shifted. Then,

$$\mathcal{L}(G, {}_3\!\Rightarrow) = \big\{x \in T^* \mid x = \varepsilon \text{ or } x = uvwb^n, uw = a^n, n \geq 0, \\ \text{and } v \text{ is defined recursively as } x\big\}.$$

For example, the string $aaaababbabbb$ is generated by G in the following way:

$$\begin{aligned} S\ {}_3\!\Rightarrow A\$\ {}_3\!\Rightarrow aAb\$\ {}_3\!\Rightarrow aaAbb\$\ {}_3\!\Rightarrow aaaAbbb\$\ {}_3\!\Rightarrow aaAabbb\$ \\ {}_3\!\Rightarrow aaaAbabbb\$\ {}_3\!\Rightarrow aaaaAbbabbb\$\ {}_3\!\Rightarrow aaaAabbabbb\$ \\ {}_3\!\Rightarrow aaaaAbabbabbb\$\ {}_3\!\Rightarrow aaaababbabbb\$\ {}_3\!\Rightarrow aaaababbabbb \end{aligned}$$

Theorem 3.7. $SC({}_3\!\Rightarrow) = \mathbf{RE}$.

Proof:
Clearly $SC({}_3\!\Rightarrow) \subseteq \mathbf{RE}$, so we only need to prove $\mathbf{RE} \subseteq SC({}_3\!\Rightarrow)$.

Let $G = (V, \Sigma, P, S)$ be the SCG constructed in the proof of Theorem 3.3. Next, we modify G to a new SCG G' satisfying $\mathcal{L}(G, {}_1\!\Rightarrow) = \mathcal{L}(G', {}_1\!\Rightarrow)$. Finally, we prove $\mathcal{L}(G', {}_3\!\Rightarrow) = \mathcal{L}(G', {}_1\!\Rightarrow)$.

Construction. Let $G' = (V, \Sigma, P', S)$ be SCG with P' containing

(1) $(S) \rightarrow (S_1 11\$\$11 S_2)$;
(2) $(A) \rightarrow (C_i(w))$ for $A \rightarrow w \in P_i$, where $i = 1, 2$;
(3) $(a, \$, \$, a) \rightarrow (\$, \varepsilon, \varepsilon, \$)$, for $a = 0, 1$;
(4) $(\$) \rightarrow (\varepsilon)$.

We establish the proof of Theorem 3.7 by demonstrating the following two claims.

Claim 6. $\mathcal{L}(G', {}_1\!\Rightarrow) = \mathcal{L}(G, {}_1\!\Rightarrow)$.

Proof:
G' is closely related to G, only the rules (1) and (3) are slightly modified. As a result the correspondence of the sentences generated by the simulated G_1, G_2, respectively, is not checked in the direction from the outermost to the central symbols but from the central to the outermost symbols. Again, if the current two symbols do not match, they cannot be erased both and the derivation blocks. □

Claim 7. $\mathcal{L}(G', {}_3\Rightarrow) = \mathcal{L}(G', {}_1\Rightarrow)$.

Proof:
Without any loss of generality, we can suppose the rules (1) and (2) are used only before the first usage of the rule (3). The context-free rules work unchanged with ${}_3\Rightarrow$. Then, for every derivation

$$S \ {}_1\Rightarrow^* \ w = w_1 11\$\$11 w_2$$

generated only by the rules (1) and (2), where $\mathrm{alph}(w) \cap (N_1 \cup N_2) = \emptyset$, there is the identical derivation

$$S \ {}_3\Rightarrow^* \ w$$

and vice versa. Since

$$w \ {}_1\Rightarrow^* \ w', w' \in \Sigma^*$$

if and only if $t'(w_1) = \mathrm{reversal}(w_2)$, we can complete the proof of the previous claim by the following one.

Claim 8. Let the sentential form w be generated only by the rules (1) and (2). Without any loss of generality, suppose $\mathrm{alph}(w) \cap (N_1 \cup N_2) = \emptyset$. Consider

$$S \ {}_3\Rightarrow^* \ w = w_1 11\$\$11 w_2.$$

Then, $w \ {}_3\Rightarrow^* \ w'$, where $w' \in \Sigma^*$ if and only if $t'(w_1) = \mathrm{reversal}(w_2)$.

For better readability, in the next proof we omit all symbols of w_1 from Σ—we consider only nonterminal symbols, which are to be erased.

Basic idea. The rules (3) are the only with 0s and 1s on their left-hand sides. These symbols are simultaneously erasing to the left and to the right of \$s checking the equality. While proceeding from the center to the edges, when there is any symbol skipped, which is remaining between \$s, there is no way, how to erase it, and no terminal string can be generated.

Consider ${}_3\Rightarrow$. Even when the symbols are erasing one after another from the center to the left and right, ${}_3\Rightarrow$ can potentially shift the left \$ to the left and the right \$ to the right skipping some symbols. Also in this case the symbols between \$s cannot be erased anymore.

Proof:
If. Recall

$$w = 10^{m_1}110^{m_2}1 \dots 10^{m_l}111\$\$1110^{m_l}1 \dots 10^{m_2}110^{m_1}1.$$

Suppose the check works properly not skipping any symbol. Then,

$$w \ {}_3\Rightarrow^* \ w' = \$\$$$

and twice applying the rule (4) the derivation finishes. □

Proof:
Only if. If $w_1 \neq \text{reversal}(w_2)$, though the check works properly,

$$w \ {}_1\!\Rightarrow^* w' = w'_1 x\$\$x' w'_2$$

and $x, x' \in \{0, 1\}$, $x \neq x'$. Continuing the check with application of the rules (3) will definitely skip x or x'. Consequently, no terminal string can be generated.

We showed that G' can generate the terminal string from the sentential form w if and only if $t'(w_1) = \text{reversal}(w_2)$, and the claim holds. □

Since $S \ {}_1\!\Rightarrow^* w$, $w \in \Sigma^*$ if and only if $S \ {}_3\!\Rightarrow^* w$, Claim 7 holds. □

We proved $\mathcal{L}(G, {}_1\!\Rightarrow) = L$, $\mathcal{L}(G', {}_1\!\Rightarrow) = \mathcal{L}(G, {}_1\!\Rightarrow)$, $\mathcal{L}(G', {}_3\!\Rightarrow) = \mathcal{L}(G', {}_1\!\Rightarrow)$; therefore, $\mathcal{L}(G', {}_3\!\Rightarrow) = L$ holds. Thus, the proof of Theorem 3.7 is completed. □

3.4. Jumping derivation mode 4

Definition 6. Let $G = (V, T, P, S)$ be an SCG. Let $uAv \in V^*$ and $(A) \rightarrow (x) \in P$. Then $uAv \ {}_4\!\Rightarrow uxv$. Let $u = u_0A_1u_1 \dots A_nu_n \in V^*$ and $(A_1,A_2,\dots,A_n) \rightarrow (x_1,x_2,\dots,x_n) \in P$, for $n \geq 2$. Then,

$$u_0A_1u_1A_2u_2 \dots u_{i-1}A_iu_iA_{i+1}u_{i+1} \dots u_{n-1}A_nu_n \ {}_4\!\Rightarrow$$
$$u_0u_1x_1u_2x_2 \dots u_{i-1}x_{i-1}u_{i_1}x_iu_{i_2}x_{i+1}u_{i_3}x_{i+2}u_{i+1} \dots x_nu_{n-1}u_n,$$

where $u_i = u_{i_1}u_{i_2}u_{i_3}$.

Informally, $v = u_0u_1x_1u_2x_2 \dots u_{i-1}x_{i-1}u_{i_1}x_iu_{i_2}x_{i+1}u_{i_3}x_{i+2}u_{i+1} \dots x_nu_{n-1}u_n$ is obtained from $u = u_0A_1u_1A_2u_2 \dots u_{i-1}A_iu_iA_{i+1}u_{i+1} \dots u_{n-1}A_nu_n$ in G by $(A_1,A_2,\dots,A_n) \rightarrow (x_1,x_2,\dots,x_n) \in P$ in ${}_4\!\Rightarrow$ as follows:

(1) $A_1, A_2, \dots, A_n$ are deleted;
(2) a central u_i is nondeterministically chosen, for some $i \in \{0, \dots, n\}$;
(3) x_i and x_{i+1} are inserted into u_i;
(4) x_j is inserted between u_j and u_{j+1}, for all $j < i$;
(5) x_k is inserted between u_{k-2} and u_{k-1}, for all $k > i + 1$.

Example 3.8. Let $G = (V, T, P, S)$, where $V = \{S, A, B, C, \$, a, b, c, d\}$, $T = \{a, b, c, d\}$, be an SCG with P containing the following rules:

(i) $(S) \rightarrow (AB\$\$BA)$ (iv) $(A, B, B, A) \rightarrow (A, C, C, A)$
(ii) $(A) \rightarrow (aAb)$ (v) $(\$, C, C, \$) \rightarrow (\varepsilon, \varepsilon, \varepsilon, \varepsilon)$
(iii) $(B) \rightarrow (cBd)$ (vi) $(A) \rightarrow (\varepsilon)$

Consider G uses ${}_4\Rightarrow$. Then, every context-sensitive rule is applied in the following way. First, all affected nonterminals are erased. Next, some position of the current sentential form called center is nondeterministically chosen. Finaly, the corresponding right-hand sides of the selected rule are inserted each at the original place of the neighbouring erased nonterminal closer to the center. The central right-hand-side strings are randomly put closer to the chosen central position. In this example, we show how to control the choice.

First the rule (i) rewrites S to $AB\$\$BA$. Then, G uses the rules (ii) and (iii) generating a sentential form

$$a^{n_1}Ab^{n_1}c^{n_2}Bd^{n_2}\$\$c^{n_3}Bd^{n_3}a^{n_4}Ab^{n_4},$$

where $n_i \geq 0$, for $i \in \{1,2,3,4\}$. If the rule (vi) is used, derivation is blocked. Next, G uses the context-sensitive rule (iv), which may act in several different ways. In any case, it inserts two Cs into the current sentential form and the only possibility to erase them is the rule (v). However, thereby we force the rule (iv) to choose the center for interchanging nonterminals between Bs and moreover to insert Cs between the two symbols \$. Finally, G continues by using the rule (ii) and eventually finishes twice the rule (vi). Consequently,

$$\mathcal{L}(G, {}_4\Rightarrow) = \big\{x \in T^* \mid x = a^{n_1}b^{n_1}c^{n_2}a^{n_3}b^{n_3}d^{n_2}c^{n_4}a^{n_5}b^{n_5}d^{n_4}a^{n_6}b^{n_6},\ n_i \geq 0, i \in \{1,2,3,4,5,6\}\big\}.$$

Then, the string $aabbcabdccddab$ is generated by G in the following way:

$$\begin{gathered}S \ {}_4\Rightarrow\ AB\$\$BA \ {}_4\Rightarrow\ aAbB\$\$BA \ {}_4\Rightarrow\ aaAbbB\$\$BA \ {}_4\Rightarrow\ aaAbbcBd\$\$BA \\ {}_4\Rightarrow\ aaAbbcBd\$\$cBdA \ {}_4\Rightarrow\ aaAbbcBd\$\$ccBddA \ {}_4\Rightarrow\ aaAbbcBd\$\$ccBddaAb \\ {}_4\Rightarrow\ aabbcAd\$CC\$ccAddab \ {}_4\Rightarrow\ aabbcAdccAddab \ {}_4\Rightarrow\ aabbcaAbdccAddab \\ {}_4\Rightarrow\ aabbcabdccAddab \ {}_4\Rightarrow\ aabbcabdccddab\end{gathered}$$

Theorem 3.9. $SC({}_4\Rightarrow) = \mathbf{RE}$.

Proof:
A obvious, $SC({}_4\Rightarrow) \subseteq \mathbf{RE}$, so we only prove $\mathbf{RE} \subseteq SC({}_4\Rightarrow)$.

Let $G = (V, \Sigma, P, S)$ be the SCG constructed in the proof of Theorem 3.3. Next, we modify G to a new SCG G' so $\mathcal{L}(G, {}_1\Rightarrow) = \mathcal{L}(G', {}_4\Rightarrow)$.

Construction. Introduce five new symbols—D,E,F,$|$, and $\top$. Set $N = \{D,E,F,|,\top\}$. Let $G' = (V', \Sigma, P', S)$ be SCG, with $V' = V \cup N$ and P' containing the rules
(1) $(S) \rightarrow (F\$S_1 11|E|11S_2\$F)$;
(2) $(A) \rightarrow (C_i(w))$ for $A \rightarrow w \in P_i$, where $i = 1, 2$;
(3) $(F) \rightarrow (FF)$;
(4) $(\$, a, a, \$) \rightarrow (\varepsilon, D, D, \varepsilon)$, for $a = 0, 1$;
(5) $(F, D, |, |, D, F) \rightarrow (\$, \varepsilon, \top, \top, \varepsilon, \$)$;
(6) $(\top, E, \top) \rightarrow (\varepsilon, |E|, \varepsilon)$;
(7) $(\$) \rightarrow (\varepsilon)$, $(E) \rightarrow (\varepsilon)$, $(|) \rightarrow (\varepsilon)$.

Claim 9. $\mathcal{L}(G, {}_1\Rightarrow) = \mathcal{L}(G', {}_4\Rightarrow)$.

Proof:
The behaviour of context-free rules remains unchanged under ${}_4\Rightarrow$. Since the rules of G' simulating the derivations of G_1 and G_2 are identical to the ones of G simulating both grammars, for every derivation of G

$$S \ {}_1\Rightarrow^* \ \$w_1 1111 w_2\$ = w,$$

where w is generated only by using the rules (1) and(2) and $\mathrm{alph}(w) \cap (N_1 \cup N_2) = \emptyset$, there is

$$S \ {}_4\Rightarrow^* \ F\$w_1 11|E|11 w_2\$\#\#F = w'$$

in G', generated by the corresponding rules (1) and (2), and vice versa. Without any loss of generality, we can consider such a sentential form in every successful derivation. Additionally, in G

$$w \ {}_1\Rightarrow^* \ v, v \in \Sigma^*$$

if and only if $t'(w_1) = \mathrm{reversal}(w_2)$; then $v = t(w)$. Therefore, we have to prove

$$w' \ {}_4\Rightarrow^* \ v', v' \in \Sigma^*$$

if and only if $t'(w_1) = \mathrm{reversal}(w_2)$. Then obviously $v' = v$ and we can complete the proof by the following claim.

Claim 10. In G', for

$$S \ {}_4\Rightarrow^* \ w = F^{i_1}\$w_1 11|E|11 w_2\$\#\#F^{i_2}, \ \mathrm{alph}(w) \cap (N_1 \cup N_2) = \emptyset,$$

where w is generated only by using the rules (1) and (2),

$$w \ {}_4\Rightarrow^* \ w',$$

where $w' \in \Sigma^*$ if and only if $t'(w_1) = \mathrm{reversal}(w_2)$, for some $i_1, i_2 \geq 0$.

The new rule (3) potentially arbitrarily multiplies the number of Fs to the left and right. Then, Fs from both sequences are simultaneously erasing by using the rule (5). Thus, without any loss of generality, suppose $i_1 = i_2$ equal the number of future usages of the rule (5).

For the sake of readability, in the next proof, in w_1, we omit all symbols from Σ—we consider only nonterminal symbols, which are to be erased.

Proof:
If. Suppose $w_1 = \mathrm{reversal}(w_2)$, then $w \ {}_4\Rightarrow^* \ \varepsilon$. We prove this by the induction on the length of w_1,w_2, where $|w_1| = |w_2| = k$.

Basis. Let $k = 0$. Then, $w = FF\$11|E|11\$FF$. Except the rules (7), the rule (4) is the only applicable. The center for interchanging the right-hand-side strings must be chosen between the two rewritten 1s and additionally inserted Ds must remain on the different sides of the central string $|E|$. Moreover, if any 1 stays outside the two Ds, it cannot be erased, so

$$FF\$11|E|11\$FF \ {}_4\Rightarrow \ FFD1|E|1DFF$$

Next, the rule (5) rewrites Ds back to \$s, erases Fs, and changes $|$s to $\top$s. The center must be chosen between the two $|$s and inserted $\top$s may not be shifted, otherwise they appear on the same side of E and the rule (6) is inapplicable. It secures the former usage of the rule (4) was as expected, so

$$FFD1|E|1DFF_4\!\Rightarrow F\$1\top E\top 1\$F$$

By the rule (6) the symbols $\top$ may be rewritten back to $|$s. No left-hand-side string may be shifted during the application of the rule and the choice of the central position has no influence, because the neighbouring symbols are rewritten. It secures the former usage of the rule (5) was as expected; therefore,

$$F\$1\top E\top 1\$F_4\!\Rightarrow F\$1|E|1\$F$$

Then, the same sequence of rules with the same restrictions can be used again to erase remaining 1s and the check is finished by the rules (7) as

$$F\$1|E|1\$F_4\!\Rightarrow FD|E|DF_4\!\Rightarrow \$\top E\top\$_4\!\Rightarrow \$|E|\$_4\!\Rightarrow^* \varepsilon$$

and the basis holds.

Induction Hypothesis. Suppose there exists $k \geq 0$ such that the claim holds for all $0 \leq m \leq k$, where

$$w = F^{i_1}\$w_1 11|E|11w_2\$F^{i_2}, |w_1| = |w_2| = m.$$

Induction Step. Consider G' generating w with

$$w = F^{i_1}\$w_1 11|E|11w_2\$F^{i_2},$$

where $|w_1| = |w_2| = k+1$, $w_1 = \text{reversal}(w_2) = aw_1'$, and $a \in \{0, 1\}$. Except the rules (7), the rule (4) is the only applicable. The center for interchanging of the right-hand-side strings must be chosen between the two rewritten 0s or 1s and additionally inserted Ds must remain on the different sides of the central string $|E|$. Moreover, the outermost 0s or 1s must be rewritten and Ds may not be shifted between the new outermost ones, otherwise they cannot be erased.

$$F^{i_1}\$w_1 11|E|11w_2\$F^{i_2}{}_4\!\Rightarrow F^{i_1}Dw_1'11|E|11w_2'DF^{i_2}$$

Next, the rule (5) rewrites Ds back to \$s, erases Fs, and changes $|$s to $\top$s. The center must be chosen between the two $|$s and inserted $\top$s may not be shifted, otherwise they appear on the same side of E and the rule (6) is inapplicable. It secures the former usage of the rule (4) was as expected.

$$F^{i_1}Dw_1'11|E|11w_2'DF^{i_2}{}_4\!\Rightarrow F^{i_1'}\$w_1'11\top E\top 11w_2'\$F^{i_2'}$$

By the rule (6) the symbols $\top$ may be rewritten back to $|$s. No left-hand-side string may be shifted during the application of the rule and the position of the chosen center has no influence, because the neighbouring symbols are rewritten. It secures the former usage of the rule (5) was as expected.

$$F^{i_1'}\$w_1'11\top E\top 11w_2'\$F^{i_2'}{}_4\!\Rightarrow F^{i_1'}\$w_1'11|E|11w_2'\$F^{i_2'} = w'$$

By induction hypothesis $w'{}_4{\Rightarrow}^* \varepsilon$, which completes the proof.

Only if. Suppose $w_1 \neq \text{reversal}(w_2)$; there is no w', where $w\ {}_4{\Rightarrow}^*\ w'$ and $w' = \varepsilon$.

Since $w_1 \neq \text{reversal}(w_2)$, $w_1 = vau$, $w_2 = u'a'v$, and $a \neq a'$. Suppose both vs are correctly erased and no symbol is skipped producing the sentential form

$$F^{i_1}\$au11|E|11u'a'\$F^{i_2}.$$

Next, the rule (4) can be applied to erase outermost 0s or 1s. However, then, there is 0 or 1 outside inserted Ds and, thus, unable to be erased, which completes the proof.

We showed that G' can generate the terminal string from the sentential form w if and only if $t'(w_1) = \text{reversal}(w_2)$, and the claim holds. □

We proved that for some $w \in \Sigma^*$, $S\ {}_1{\Rightarrow}^*\ w$ in G if and only if $S\ {}_4{\Rightarrow}^*\ w$ in G', and the claim holds. □

Since $\mathcal{L}(G, {}_1{\Rightarrow}) = \mathcal{L}(G', {}_4{\Rightarrow}) = L$, the proof of Theorem 3.9 is completed. □

3.5. Jumping derivation mode 5

Definition 7. Let $G = (V, T, P, S)$ be an SCG. Let $uAv \in V^*$ and $(A) \to (x) \in P$. Then $uAv\ {}_5{\Rightarrow}\ uxv$. Let $u = u_0A_1u_1 \ldots A_nu_n \in V^*$ and $(A_1,A_2,\ldots,A_n) \to (x_1,x_2,\ldots,x_n) \in P$, for $n \geq 2$. Then,

$$u_0A_1u_1A_2 \ldots u_{i-1}A_{i-1}u_iA_iu_{i+1} \ldots A_nu_n\ {}_5{\Rightarrow}$$
$$u_{0_1}x_1u_{0_2}x_2u_1 \ldots x_{i-1}u_{i-1}u_iu_{i+1}x_i \ldots u_{n_1}x_nu_{n_2},$$

where $u_0 = u_{0_1}u_{0_2}$, $u_n = u_{n_1}u_{n_2}$.

Informally, scattered context grammar G obtains $u_{0_1}x_1u_{0_2}x_2u_1 \ldots x_{i-1}u_{i-1}u_iu_{i+1}x_i \ldots u_{n_1}x_nu_{n_2}$ from $u_0A_1u_1A_2 \ldots u_{i-1}A_{i-1}u_iA_iu_{i+1} \ldots A_nu_n$ by $(A_1,A_2,\ldots,A_n) \to (x_1,x_2,\ldots,x_n) \in P$ in ${}_5{\Rightarrow}$ as follows:

(1) $A_1, A_2, \ldots, A_n$ are deleted;
(2) a central u_i is nondeterministically chosen, for some $i \in \{0, \ldots, n\}$;
(3) x_1 and x_n are inserted into u_0 and u_n, respectively;
(4) x_j is inserted between u_{j-2} and u_{j-1}, for all $1 < j \leq i$;
(5) x_k is inserted between u_k and u_{k+1}, for all $i + 1 \leq k < n$.

Example 3.10. Let $G = (V, T, P, S)$, where $V = \{S, A, B, \$, a, b\}$, $T = \{a, b\}$, be an SCG with P containing the following rules:

(i) $(S) \to (\$AA\$)$
(ii) $(A) \to (aAb)$
(iii) $(A, A) \to (B, B)$
(iv) $(B, \$, \$, B) \to (A, \varepsilon, \varepsilon, A)$
(v) $(A) \to (\varepsilon)$

Recall Example 3.8. ${}_4\Rightarrow$ interchanges the positions of nonterminals influenced by context-sensitive rules in the direction from the outer ones to the central ones. Opposed to ${}_4\Rightarrow$, ${}_5\Rightarrow$ interchanges nonterminals in the direction from a nondeterministically chosen center. In the present example, we show one possibility to control the choice.

Consider G uses ${}_5\Rightarrow$. First the rule (i) rewrites S to $\$AA\$$. Then, G uses the rule (ii) generating the sentential form

$$\$a^m A b^m a^n A b^n \$,$$

where $m, n \geq 0$. If the rule (v) is used, derivation is blocked, because there is no way to erase the symbols $\$$. Next, G uses the context-sensitive rule (iii), which nondeterministically choses a center and nondeterministically shifts Bs from the previous positions of As in the direction from this center. However, for the future application of the rule (iv) the chosen center must lie between As and moreover Bs must be inserted as the leftmost and the rightmost symbols of the current sentential form. The subsequent usage of the rule (iv) preserves As as the leftmost and the rightmost symbols independently of the effect of ${}_5\Rightarrow$. Finally, G continues by using the rule (ii) and eventually finishes twice the rule (v). If the rule (iii) is used again, there is no possibility to erase inserted Bs. Consequently,

$$\mathcal{L}(G, {}_5\Rightarrow) = \left\{x \in T^* \mid x = a^k b^k a^l b^l a^m b^m a^n b^n, k, l, m, n \geq 0\right\}.$$

Then, the string $aabbabaaabbb$ is generated by G in the following way:

$$\begin{gathered} S \;{}_5\Rightarrow \$AA\$ \;{}_5\Rightarrow \$aAbA\$ \;{}_5\Rightarrow \$aaAbbA\$ \;{}_5\Rightarrow \$aaAbbaAb\$ \\ {}_5\Rightarrow B\$aabbab\$B \;{}_5\Rightarrow AaabbabA \;{}_5\Rightarrow AaabbabaAb \;{}_5\Rightarrow AaabbabaaAbb \\ {}_5\Rightarrow AaabbabaaaAbbb \;{}_5\Rightarrow aabbabaaaAbbb \;{}_5\Rightarrow aabbabaaabbb \end{gathered}$$

Theorem 3.11. $SC({}_5\Rightarrow) = \mathbf{RE}$.

Proof:
As obvious, $SC({}_5\Rightarrow) \subseteq \mathbf{RE}$, so we only prove $\mathbf{RE} \subseteq SC({}_5\Rightarrow)$.

Let $G = (V, \Sigma, P, S)$ be the SCG constructed in the proof of Theorem 3.3. Next, we modify G to a new SCG G' so $\mathcal{L}(G, {}_1\Rightarrow) = \mathcal{L}(G', {}_5\Rightarrow)$.

Construction. Introduce four new symbols—D,E,F, and $\circ$. Set $N = \{D,E,F,\circ\}$. Let $G' = (V', \Sigma, P', S)$ be SCG, with $V' = V \cup N$ and P' containing the rules

(1) $(S) \to (\$S_1 1111 S_2\$ \circ E \circ F)$;
(2) $(A) \to (C_i(w))$ for $A \to w \in P_i$, where $i = 1, 2$;
(3) $(F) \to (FF)$;
(4) $(\$, a, a, \$, E, F) \to (\varepsilon, \varepsilon, \$, \$, \varepsilon, D)$, for $a = 0, 1$;
(5) $(\circ, D, \circ) \to (\varepsilon, \circ E \circ, \varepsilon)$;
(6) $(\$) \to (\varepsilon)$, $(E) \to (\varepsilon)$, $(\circ) \to (\varepsilon)$.

Claim 11. $\mathcal{L}(G, {}_1\Rightarrow) = \mathcal{L}(G', {}_5\Rightarrow)$.

Proof:
Context-free rules are not influenced by ${}_5\Rightarrow$. The rule (3) must generate precisely as many Fs as the number of applications of the rule (4). Context-sensitive rules of G' correspond to context-sensitive rules of G, except the special rule (5). We show, the construction of G' forces context-sensitive rules to work exactly in the same way as the rules of G do.

Every application of the rule (4) must be followed by the application of the rule (5) to rewrite D back to E, which requires the symbol D between two $\circ$s. It ensures the previous usage of context-sensitive rule selected the center to the right of the rightmost affected nonterminal and all right-hand-side strings changed their positions with the more left ones. The leftmost right-hand-side string is then shifted randomly to the left, but it is always ε. ${}_5\Rightarrow$ has no influence on the rule (5).

From the construction of G', it works exactly in the same way as G does. □

$\mathcal{L}(G, {}_1\Rightarrow) = \mathcal{L}(G', {}_5\Rightarrow)$ and $\mathcal{L}(G, {}_1\Rightarrow) = L$; therefore $\mathcal{L}(G', {}_5\Rightarrow) = L$. Thus, the proof of Theorem 3.11 is completed. □

3.6. Jumping derivation mode 6

Definition 8. Let $G = (V, T, P, S)$ be an SCG. Let $uAv \in V^*$ and $(A) \to (x) \in P$. Then $uAv \;{}_6\Rightarrow\; uxv$. Let $u = u_0A_1u_1 \dots A_nu_n \in V^*$ and $(A_1,A_2,\dots,A_n) \to (x_1,x_2,\dots,x_n) \in P$, for $n \geq 2$. Then,

$$u_0A_1u_1A_2u_2 \dots u_{i-1}A_iu_iA_{i+1}u_{i+1} \dots u_{n-1}A_nu_n \;{}_6\Rightarrow$$
$$u_0u_1x_1u_2x_2 \dots u_{i-1}x_{i-1}u_ix_{i+2}u_{i+1} \dots x_nu_{n-1}u_n.$$

Informally, scattered context grammar G obtains $u_0u_1x_1u_2x_2 \dots u_{i-1}x_{i-1}u_ix_{i+2}u_{i+1} \dots x_nu_{n-1}u_n$ from $u_0A_1u_1A_2u_2 \dots u_{i-1}A_iu_iA_{i+1}u_{i+1} \dots u_{n-1}A_nu_n$ by using $(A_1,A_2,\dots,A_n) \to (x_1, x_2,\dots,x_n) \in P$ in ${}_6\Rightarrow$ as follows:

(1) $A_1, A_2, \dots, A_n$ are deleted;
(2) a central u_i is nondeterministically chosen, for some $i \in \{0, \dots, n\}$;
(3) x_j is inserted between u_j and u_{j+1}, for all $j < i$;
(4) x_k is inserted between u_{k-2} and u_{k-1}, for all $k > i + 1$.

Example 3.12. Let $G = (V, T, P, S)$, where $V = \{S, A, B, a, b\}$, $T = \{a, b\}$, be an SCG with P containing the following rules:

(i) $(S) \to (ABBA)$	(iii) $(A, B, B, A) \to (AB, B, B, BA)$
(ii) $(A) \to (aAb)$	(iv) $(A, B, B, A) \to (\varepsilon, B, B, \varepsilon)$

Consider G uses ${}_6\Rightarrow$. ${}_6\Rightarrow$ interchanges nonterminals similarly as ${}_4\Rightarrow$ does in Example 3.8, however, the central nonterminals are removed. We can use this property to determine, which are the central ones; demonstration follows.

The rules (i) and (ii) are context-free, not affected by ${}_6\Rightarrow$. First the starting rule (i) rewrites S to $ABBA$. Then, G uses the rule (ii) generating the sentential form

$$a^m Ab^m BBa^n Ab^n,$$

where $m, n \geq 0$. Next, G uses the context-sensitive rule (iii) or (iv). Notice, there is no rule erasing Bs, thus in both cases the center of interchanging of nonterminals must be chosen between the two Bs. Otherwise, in both cases there is exactly one A remaining, thus the only applicable rule is the rule (ii), which is context-free and not erasing. Therefore, G uses the rule (iii) generating the sentential form

$$a^m b^m ABBAa^n b^n$$

and continues by using the rule (ii) or it uses the rule (iv) and finishes the derivation.

Subsequently, the language G generates is

$$\mathcal{L}(G, {}_6\Rightarrow) = \{x \in T^* \mid x = a^{n_1} b^{n_1} a^{n_2} b^{n_2} \dots a^{n_{2k}} b^{n_{2k}}, k, n_i \geq 0, 1 \leq i \leq 2k\}.$$

Then, the string $aabbabaabbab$ is generated by G in the following way:

$$\begin{gathered} S \,{}_6\Rightarrow\, ABBA \,{}_6\Rightarrow\, aAbBBA \,{}_6\Rightarrow\, aaAbbBBA \,{}_6\Rightarrow\, aaAbbBBaAb \\ {}_6\Rightarrow\, aabbABBAab \,{}_6\Rightarrow\, aabbaAbBBAab \,{}_6\Rightarrow\, aabbaAbBBaAbab \\ {}_6\Rightarrow\, aabbaAbBBaaAbbab \,{}_6\Rightarrow\, aabbabaabbab \end{gathered}$$

Theorem 3.13. $SC({}_6\Rightarrow) = \mathbf{RE}$.

Proof:
Clearly, $SC({}_6\Rightarrow) \subseteq \mathbf{RE}$. Next, we prove $\mathbf{RE} \subseteq SC({}_6\Rightarrow)$.

Let $G = (V, \Sigma, P, S)$ be the SCG constructed in the proof of Theorem 3.3. Next, we modify G to a new SCG G' so $\mathcal{L}(G, {}_1\Rightarrow) = \mathcal{L}(G', {}_6\Rightarrow)$.

Construction. Introduce two new symbols—E and F. Let $G' = (V', \Sigma, P', S)$ be SCG, with $V' = V \cup \{E, F\}$ and P' containing the rules

(1) $(S) \rightarrow (F\$S_1 1111 S_2\$)$;
(2) $(A) \rightarrow (C_i(w))$ for $A \rightarrow w \in P_i$, where $i = 1, 2$;
(3) $(F) \rightarrow (FF)$;
(4) $(F, \$, a, a, \$) \rightarrow (E, E, \varepsilon, \$, \$)$, for $a = 0, 1$;
(5) $(\$) \rightarrow (\varepsilon)$.

Claim 12. $\mathcal{L}(G, {}_1\Rightarrow) = \mathcal{L}(G', {}_6\Rightarrow)$.

Proof:
Context-free rules are not influenced by ${}_6\Rightarrow$. Context-sensitive rules of G' closely correspond to context-sensitive rules of G. The new symbols are used to force modified rules to act in the same way as sample ones do. The symbols F are first multiplied and then consumed by context-sensitive rules,

so their number must equal the number of usages of these rules. The new symbols E are essential. E never appears on the left-hand side of any rule, thus whenever it is inserted into the sentential form, no terminal string can be generated. Therefore, the center is always chosen between two Es, which are basically never inserted, and other right-hand-side strings are then inserted deterministically.

G' with ${}_6\Rightarrow$ works in the same way as G with ${}_1\Rightarrow$ does. □

$\mathcal{L}(G, {}_1\Rightarrow) = \mathcal{L}(G', {}_6\Rightarrow)$, hence $\mathcal{L}(G', {}_6\Rightarrow) = L$. Thus, the proof of Theorem 3.13 is completed. □

3.7. Jumping derivation mode 7

Definition 9. Let $G = (V, T, P, S)$ be an SCG. Let $(A) \rightarrow (x) \in P$ and $uAv \in V^*$. Then $uAv \ {}_7\Rightarrow \ uxv$. Let $u = u_0A_1u_1 \dots A_nu_n \in V^*$ and $(A_1,A_2,\dots,A_n) \rightarrow (x_1,x_2,\dots,x_n) \in P$, for $n \geq 2$. Then,

$$u_0A_1u_1A_2 \dots u_{i-1}A_iu_iA_{i+1}u_{i+1} \dots A_nu_n \ {}_7\Rightarrow$$
$$u_0x_2u_1 \dots x_iu_{i-1}u_iu_{i+1}x_{i+1} \dots u_n.$$

Informally, G obtains $u_0x_2u_1 \dots x_iu_{i-1}u_iu_{i+1}x_{i+1} \dots u_n$ from $u_0A_1u_1A_2 \dots u_{i-1}A_iu_iA_{i+1}u_{i+1} \dots A_nu_n$ by using $(A_1,A_2, \dots, A_n) \rightarrow (x_1,x_2,\dots,x_n) \in P$ in ${}_7\Rightarrow$ as follows:

(1) $A_1, A_2, \dots, A_n$ are deleted;
(2) a central u_i is nondeterministically chosen, for some $i \in \{0 \dots, n\}$;
(3) x_j is inserted between u_{j-2} and u_{j-1}, for all $1 < j \leq i$;
(4) x_k is inserted between u_k and u_{k+1}, for all $i + 1 \leq k < n$.

Example 3.14. Let $G = (V, T, P, S)$, where $V = \{S, A, B, C, \$, a, b, c\}$, $T = \{a, b, c\}$, be an SCG with P containing the following rules:

(i) $(S) \rightarrow (ABC\$)$
(ii) $(A) \rightarrow (aAa)$
(iii) $(B) \rightarrow (bBb)$
(iv) $(C) \rightarrow (cCc)$
(v) $(A, B, C) \rightarrow (A, B, C)$
(vi) $(A, B) \rightarrow (A, B)$
(vii) $(A, \$) \rightarrow (\varepsilon, \varepsilon)$

Consider G uses ${}_7\Rightarrow$. ${}_7\Rightarrow$ interchanges nonterminals in the direction from the nondeterministically chosen center and erases the outermost nonterminals. In this example, we show that we may force the center to lie outside the part of a sentential form between the affected nonterminals.

The derivation starts by using the starting rule (i) and continues by using the rules (ii) through (iv) generating the sentential form

$$a^mAa^mb^nBb^nc^lCc^l\$,$$

where $m, n, l \geq 0$. Next, G uses the context-sensitive rule (v) choosing the center to the left of A erasing C. If a different central position is chosen, the symbol A is erased while B or C cannot be erased in the future and the derivation is blocked. There is the same situation, if one of the rules (vi) or (vii) is used instead. Notice, no rule erases B or C. Then, the derivation continues by using the rules (ii) and (iii) and eventually the rule (vi) rewriting B to A and erasing B. Otherwise, A is erased and the symbol \$ cannot be erased any more. G continues by using the rule (ii) and finally finishes the derivation with the rule (vii). Subsequently,

$$\mathcal{L}(G, {}_7\Rightarrow) = \big\{x \in T^* \mid x = a^{2m_1}b^{n_1}a^{2m_2}b^{n_1}c^l b^{n_2}a^{2m_3}b^{n_2}c^l,\ m_1, m_2, m_3, n_1, n_2, l \geq 0\big\}.$$

Then, the string $aabaabccbaabcc$ is generated by G in the following way:

$$S \;{}_7\Rightarrow\; ABC\$ \;{}_7\Rightarrow\; aAaBC\$ \;{}_7\Rightarrow\; aAabBbC\$ \;{}_7\Rightarrow\; aAabBbcCc\$$$
$${}_7\Rightarrow\; aAabBbccCcc\$ \;{}_7\Rightarrow\; aabAbccBcc\$ \;{}_7\Rightarrow\; aabaAabccBcc\$ \;{}_7\Rightarrow\; aabaAabccbBbcc\$$$
$${}_7\Rightarrow\; aabaabccbAbcc\$ \;{}_7\Rightarrow\; aabaabccbaAabcc\$ \;{}_7\Rightarrow\; aabaabccbaabcc$$

Theorem 3.15. $SC({}_7\Rightarrow) = \mathbf{RE}$.

Proof:
Clearly, $SC({}_7\Rightarrow) \subseteq \mathbf{RE}$. We prove $\mathbf{RE} \subseteq SC({}_7\Rightarrow)$.

Let $G = (V, \Sigma, P, S)$ be the SCG constructed in the proof of Theorem 3.3. Next, we modify G to a new SCG G' so $\mathcal{L}(G, {}_1\Rightarrow) = \mathcal{L}(G', {}_7\Rightarrow)$.

Construction. Introduce four new symbols—E,F,H, and $|$. Set $N = \{E, F, H, |\}$. Let $G' = (V', \Sigma, P', S)$ be SCG, with $V' = V \cup N$ and P' containing the rules

(1) $(S) \to (FHS_1 11\$|\$11S_2)$;
(2) $(A) \to (C_i(w))$ for $A \to w \in P_i$, where $i = 1, 2$;
(3) $(F) \to (FF)$;
(4) $(a, \$, \$, a) \to (\varepsilon, E, E, \varepsilon)$, for $a = 0, 1$;
(5) $(F, H, E, |, E) \to (H, \$, |, \$, \varepsilon)$;
(6) $(\$) \to (\varepsilon)$, $(H) \to (\varepsilon)$, $(|) \to (\varepsilon)$.

Claim 13. $\mathcal{L}(G, {}_1\Rightarrow) = \mathcal{L}(G', {}_7\Rightarrow)$.

Proof:
The behaviour of context-free rules remains unchanged under ${}_7\Rightarrow$. Since the rules of G' simulating the derivations of G_1, G_2, respectively, are identical to the ones of G simulating both grammars, for every derivation of G

$$S \;{}_1\Rightarrow^*\; \$w_1 1111 w_2\$ = w,$$

where w is generated only by using the rules (1) and(2) and $\mathrm{alph}(w) \cap (N_1 \cup N_2) = \emptyset$, there is

$$S \;{}_7\Rightarrow^*\; FHw_1 11\$|\$11w_2 = w'$$

in G', generated by the corresponding rules (1) and (2), and vice versa. Without any loss of generality, we can consider such a sentential form in every successful derivation. Additionally, in G

$$w \;{}_1\Rightarrow^*\; v, v \in \Sigma^*$$

if and only if $t'(w_1) = \text{reversal}(w_2)$; then $v = t(w)$. Therefore, we have to prove

$$w' \ {}_4\Rightarrow^* \ v', v' \in \Sigma^*$$

if and only if $t'(w_1) = \text{reversal}(w_2)$. Then obviously $v' = v$ and we can complete the proof by the following claim.

Claim 14. In G', for some $i \geq 1$,

$$S \ {}_7\Rightarrow^* \ w = F^i H w_1 \$|\$ w_2 E,$$

where w is generated only by using the rules (1) through (3) and $\text{alph}(w) \cap (N_1 \cup N_2) = \emptyset$. Then,

$$w \ {}_7\Rightarrow^* \ w',$$

where $w' \in \Sigma^*$ if and only if $t'(w_1) = \text{reversal}(w_2)$.

The new rule (3) may potentially arbitrarily multiply the number of Fs to the left. Then, Fs are erasing by using the rule (5). Thus, without any loss of generality, suppose i equals the number of the future usages of the rule (5).

For the sake of readability, in the next proof we omit all symbols in w_1 from Σ—we consider only nonterminal symbols, which are to be erased.

Proof:
If. Suppose $w_1 = \text{reversal}(w_2)$, then $w \ {}_7\Rightarrow^* \ \varepsilon$. We prove this by the induction on the length of w_1, w_2, where $|w_1| = |w_2| = k$. Then obviously $i = k$. By the construction of G', the least k equals 2, but we prove the claim for all $k \geq 0$.

Basis. Let $k = 0$. Then,

$$w = H\$|\$.$$

By the rules (6)

$$H\$|\$ \ {}_7\Rightarrow^* \ \varepsilon$$

and the basis holds.

Induction Hypothesis. Suppose there exists $k \geq 0$ such that the claim holds for all m, where

$$w = F^m H w_1 \$|\$ w_2, |w_1| = |w_2| = m, 0 \leq m \leq k.$$

Induction Step. Consider G' generates w, where

$$w = F^{k+1} H w_1 \$|\$ w_2, |w_1| = |w_2| = k + 1.$$

Since $w_1 = \text{reversal}(w_2)$ and $|w_1| = |w_2| = k + 1$, $w_1 = w_1' a$, $w_2 = a w_2'$. The symbols a can be erased by application of the rules (4) and (5) under several conditions. First, when the rule (4) is applied, the center for interchanging right-hand-side strings must be chosen between the two \$s, otherwise both Es appear on the same side of the symbol | and the rule (5) is not applicable. Next, no 0 or 1 may be skipped, while proceeding in the direction from the center to the edges. Finally, when the rule (5) is applied, a center must be chosen to the left of F, otherwise H is erased and the future application of this rule is excluded.

$$F^{k+1}Hw'_1a\$|\$aw'_2 \;{}_7\!\Rightarrow\; F^{k+1}Hw'_1D|Dw'_2 \;{}_7\!\Rightarrow\; F^kHw'_1\$|\$w'_2 = w'$$

By induction hypothesis, $w' {}_7\!\Rightarrow^* \varepsilon$, which completes the proof.

Only if. Suppose $w_1 \neq \text{reversal}(w_2)$, then, there is no w', where $w \;{}_7\!\Rightarrow^*\; w'$ and $w' = \varepsilon$.

Since $w_1 \neq \text{reversal}(w_2)$, $w_1 = uav$, $w_2 = va'u'$, and $a \neq a'$. Suppose both vs are correctly erased and no symbol is skipped producing the sentential form

$$F^iHua\$|\$a'u'.$$

Next the rule (4) can be applied to erase innermost 0s or 1s. However, since $a \neq a'$, even if the center is chosen properly between the two \$s, there is 0 or 1 between inserted Es and, thus, unable to be erased, which completes the proof.

We showed that G' can generate the terminal string from the sentential form w if and only if $t'(w_1) = \text{reversal}(w_2)$, and the claim holds. □

We proved $S \;{}_1\!\Rightarrow^*\; w$, $w \in \Sigma^*$, in G if and only if $S \;{}_7\!\Rightarrow^*\; w$ in G', hence $\mathcal{L}(G, {}_1\!\Rightarrow) = \mathcal{L}(G', {}_7\!\Rightarrow)$ and the claim holds. □

Since $\mathcal{L}(G, {}_1\!\Rightarrow) = \mathcal{L}(G', {}_7\!\Rightarrow)$ and $\mathcal{L}(G, {}_1\!\Rightarrow) = L$, the proof of Theorem 3.15 is completed. □

3.8. Jumping derivation mode 8

Definition 10. Let $G = (V, T, P, S)$ be an SCG. Let $u = u_0A_1u_1 \dots A_nu_n \in V^*$ and $(A_1, A_2, \dots, A_n) \rightarrow (x_1, x_2, \dots, x_n) \in P$, for $n \geq 1$. Then,

$$u_0A_1u_1A_2u_2 \dots A_nu_n \;{}_8\!\Rightarrow\; v_0x_1v_1x_2v_2 \dots x_nv_n,$$

where $u_0z_1 = v_0$, $z_2u_n = v_n$, $|u_0u_1 \dots u_{j-1}| \leq |v_0v_1 \dots v_j|$, $|u_{j+1} \dots u_n| \leq |v_jv_{j+1} \dots v_n|$, $0 < j < n$, and $z_1, z_2 \in V^*$.

Informally, G obtains $v_0x_1v_1x_2v_2 \dots x_nv_n$ from $u_0A_1u_1A_2u_2 \dots A_nu_n$ by using $(A_1, A_2, \dots, A_n) \rightarrow (x_1, x_2, \dots, x_n) \in P$ in ${}_8\!\Rightarrow$ as follows:

(1) $A_1, A_2, \dots, A_n$ are deleted;
(2) x_1 and x_n are inserted into u_1 and u_{n-1}, respectively;
(3) x_i is inserted into $u_{i-1}u_i$, for all $1 < i < n$, to the right of x_{i-1} and to the left of x_{i+1}.

Example 3.16. Let $G = (V, T, P, S)$, where $V = \{S, \overline{S}, A, B, C, a, b, c\}$, $T = \{a, b, c\}$, be an SCG with P containing the following rules:

(i)	$(S) \to (AS)$	(iv)	$(\overline{S}) \to (B)$
(ii)	$(S) \to (\overline{S})$	(v)	$(B) \to (BB)$
(iii)	$(\overline{S}) \to (b\overline{S}cC)$	(vi)	$(A, B, C) \to (a, \varepsilon, \varepsilon)$

Consider G uses ${}_8\!\Rightarrow$. ${}_8\!\Rightarrow$ acts in a similar way as ${}_2\!\Rightarrow$ does. When a rule is to be applied, there is a nondeterministically chosen center in between the affected nonterminals and rule right-hand-side strings can be shifted in the direction to this center, but not farther than the neighboring affected nonterminal was.

The rules (i) through (v) are context-free. Without any loss of generality, we suppose these rules are used only before the first application of the rule (vi) producing the string

$$A^m b^n B^l (cC)^n.$$

The derivation finishes with the sequence of applications of the rule (vi). For As, Bs, and Cs are being rewritten together, $m = n = l$. Moreover, inserted a is always between the rewritten A and B. Subsequently,

$$\mathcal{L}(G, {}_8\!\Rightarrow) = \{x \in T^* \mid x = wc^n, w \in \{a, b\}^*, \text{occur}(a, w) = \text{occur}(b, w) = n, n \geq 1\}.$$

For example, the string $baabbaccc$ is generated by G in the following way:

$$S \;{}_8\!\Rightarrow AS \;{}_8\!\Rightarrow AAS \;{}_8\!\Rightarrow AAAS \;{}_8\!\Rightarrow AAA\overline{S} \;{}_8\!\Rightarrow AAAb\overline{S}cC \;{}_8\!\Rightarrow AAAbb\overline{S}cCcC$$
$${}_8\!\Rightarrow AAAbbb\overline{S}cCcCcC \;{}_8\!\Rightarrow AAAbbbBcCcCcC \;{}_8\!\Rightarrow AAAbbbBBcCcCcC$$
$${}_8\!\Rightarrow AAAbbbBBBcCcCcC \;{}_8\!\Rightarrow AAbbbaBBccCcC \;{}_8\!\Rightarrow AbabbaBcccC \;{}_8\!\Rightarrow baabbaccc$$

Theorem 3.17. $SC({}_8\!\Rightarrow) = \mathbf{RE}$.

Proof:
Prove this theorem by analogy with the proof of Theorem 3.5. □

3.9. Jumping derivation mode 9

Definition 11. Let $G = (V, T, P, S)$ be an SCG. Let $u = u_0A_1u_1 \dots A_nu_n \in V^*$ and $(A_1,A_2, \dots, A_n) \to (x_1,x_2,\dots,x_n) \in P$, for $n \geq 1$. Then,

$$u_0A_1u_1A_2u_2 \dots A_nu_n \;{}_9\!\Rightarrow v_0x_1v_1x_2v_2 \dots x_nv_n,$$

where $u_0 = v_0z_1$, $u_n = z_2v_n$, $|u_0u_1 \dots u_{j-1}| \leq |v_0v_1 \dots v_j|$, $|u_{j+1} \dots u_n| \leq |v_jv_{j+1} \dots v_n|$, $0 < j < n$, and $z_1, z_2 \in V^*$.

Informally, G obtains $v_0x_1v_1x_2v_2 \dots x_nv_n$ from $u_0A_1u_1A_2u_2 \dots A_nu_n$ by using $(A_1,A_2, \dots, A_n) \to (x_1, x_2, \dots ,x_n) \in P$ in ${}_9\!\Rightarrow$ as follows:

(1) $A_1, A_2, \dots, A_n$ are deleted;
(2) x_1 and x_n are inserted into u_0 and u_n, respectively;
(3) x_i is inserted into $u_{i-1}u_i$, for all $1 < i < n$, to the right of x_{i-1} and to the left of x_{i+1}.

Example 3.18. Let $G = (V, T, P, S)$, where $V = \{S, \overline{S}, A, B, C, \$, a, b, c\}$, $T = \{a, b, c\}$, be an SCG with P containing the following rules:

$$
\begin{array}{rlrl}
\text{(i)} & (S) \to (aSa) & \text{(v)} & (C) \to (cBC\$) \\
\text{(ii)} & (S) \to (A) & \text{(vi)} & (C) \to (\varepsilon) \\
\text{(iii)} & (A) \to (\$A) & \text{(vii)} & (\$, B, \$) \to (\varepsilon, b, \varepsilon) \\
\text{(iv)} & (A) \to (C) & &
\end{array}
$$

Consider G uses ${}_9\Rightarrow$. ${}_9\Rightarrow$ acts similarly to ${}_3\Rightarrow$ with respect to the direction of shift of the rule right-hand sides, but with limitation as in ${}_8\Rightarrow$. When a rule is to be applied, there is a nondeterministically chosen center in between the affected nonterminals and rule right-hand-side strings can be shifted in the direction from this center, but not farther than the neighboring affected nonterminal was.

The rules (i) through (vi) are context-free. Without any loss of generality, we can suppose these rules are used only before the first application of the rule (vii), which produce the sentential form

$$a^m \$^n (cB)^l \$^l a^m.$$

The derivation finishes with the sequence of applications of the rule (vii). The symbols \$ and Bs are being rewritten togehter, thus $n = l$ must hold. Additionally, ${}_9\Rightarrow$ ensures, b is always inserted between the rewritten \$s. Subsequently,

$$\mathcal{L}(G, {}_9\Rightarrow) = \big\{x \in T^* \mid x = a^m w a^m, w \in \{b, c\}^*, \text{occur}(b, w) = \text{occur}(c, w), m \geq 0\big\}.$$

For example, the string $aabcbcaa$ is generated by G in the following way:

$$
\begin{array}{c}
S \;{}_9\Rightarrow\; aSa \;{}_9\Rightarrow\; aaSaa \;{}_9\Rightarrow\; aaAaa \;{}_9\Rightarrow\; aa\$Aaa \;{}_9\Rightarrow\; aa\$\$Aaa \\
{}_9\Rightarrow\; aa\$\$Caa \;{}_9\Rightarrow\; aa\$\$cBC\$aa \;{}_9\Rightarrow\; aa\$\$cBcBC\$\$aa \\
{}_9\Rightarrow\; aa\$\$cBcB\$\$aa \;{}_9\Rightarrow\; aa\$bccB\$aa \;{}_9\Rightarrow\; aabcbcaa
\end{array}
$$

Theorem 3.19. $SC({}_9\Rightarrow) = \mathbf{RE}$.

Proof:
Prove this theorem by analogy with the proof of Theorem 3.7. □

4. Future investigation

First, this final section describes a possible application of jumping scattered context grammars in terms of bioinformatics. Then, it formulates several open problems.

Applications perspectives

Primarily and principally, the present paper represents a theoretically oriented study. Nevertheless, next, we add some remarks and suggestions concerning future application-related perspectives of jumping grammars. We intentionally present all these suggestions as straightforwardly and simply as possible.

As already sketched, jumping grammars serve as grammatical models that allow us to explore information processing performed in a discontinuous way adequately and rigorously. Consequently, applications of these grammars are expected in any scientific area involving this kind of information processing, ranging from applied mathematics through computational linguistics and compiler writing up to data mining and biology-related informatics. Taking into account the way these grammars are conceptualized, we see that they are particularly useful and applicable under the circumstances that primarily concern the number of occurrences of various symbols or substrings rather than their mutual context.

To give an insight into future applications of jumping scattered context grammars in terms of bioinformatics, consider DNA computing, whose significance is indisputable in computer science at present. Recall that a DNA is a molecule encoding genetic information by a repetition of four basic units called nucleotides—namely, guanine, adenine, thymine, and cytosine, denoted by letters G, A, T, and C, respectively. In terms of formal language theory, a DNA is described as a string over $\{G, A, T, C\}$; for instance,

$$GGGGAGTGGGATTGGGAGAGGGGTTTGCCCCGCTCCC.$$

Suppose that a DNA-computing-related investigation needs to study all the strings that contain the same number of As and Cs, where all As precede Cs; for instance, $AGGAATCGCGTC$ is a proper string, but $CGCACCGGTA$ is not. Consider the jumping scattered context grammar containing rules

$$(1) \rightarrow (23), (3) \rightarrow (G3), (3) \rightarrow (T3), (3) \rightarrow (4), (2, 4) \rightarrow (A2, 4C), (2)|(4) \rightarrow (\varepsilon),$$

where 1 through 4 are nonterminal symbols with 1 being the start nonterminal, ε is the empty string, and G, A, T, and C are terminal symbols. Assume that the grammar works under ${}_2\Rightarrow$. It first generates an arbitrary string of Gs and Ts, in which there are no restrictions, by classical regular productions, since ${}_2\Rightarrow$ does not change the behaviour of context-free rules. However, then it comes the essential phase generating As and Cs. Indeed, the only context-sensitive rule under ${}_2\Rightarrow$ generates the equal number of As and Cs randomly scattered through the resulting sentence, but always with As preceding Cs. For instance, previously mentioned string $AGGAATCGCGTC$ can be generated by the following derivation.

$$\begin{array}{lll}
1 \;{}_2\Rightarrow 23 & {}_2\Rightarrow 2G3 & {}_2\Rightarrow 2GG3 \\
\;{}_2\Rightarrow 2GGT3 & {}_2\Rightarrow 2GGTG3 & {}_2\Rightarrow 2GGTGG3 \\
\;{}_2\Rightarrow 2GGTGGT3 & {}_2\Rightarrow 2GGTGGT4 & {}_2\Rightarrow A2GGTGGT4C \\
\;{}_2\Rightarrow AGGA2TG4CGTC & {}_2\Rightarrow AGGAA2T4CGCGTC & {}_2\Rightarrow^2 AGGAATCGCGTC
\end{array}$$

As obvious, under ${}_2\Rightarrow$, the grammar generates the language consisting of all the strings satisfying the above-stated requirements. Therefore, as we can see, jumping grammars may fulfil a useful role in studies related to DNA computing in the future.

Open problem areas

Finally, let us suggest some open problem areas concerning the subject of this paper.

(I) Return to derivation modes (1) through (9) in Section 1. Introduce and study further modes. For instance, in a more general way, discuss a jumping derivation mode, in which the only restriction is to preserve a mutual order of the inserted right-hand-side strings, which can be nondeterministicaly spread across the whole sentential form regardless of the positions of the rewritten nonterminals. In a more restrictive way, study a jumping derivation mode over words satisfying some prescribed requirements, such as a membership in a regular language.

(II) Consider propagating versions of jumping scattered context grammars. In other words, rule out erasing rules in them. Reconsider the investigation of the present paper in their terms.

(III) The present paper has often demonstrated that some jumping derivation modes work just like ordinary derivation modes in scattered context grammars. State general combinatorial properties that guarantee this behaviour.

(IV) Establish normal forms of scattered context grammars working in jumping ways.

Acknowledgments

The authors deeply thank both anonymous referees for useful comments and suggestions regarding the first version of this paper. This work was supported by the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070), the TAČR grant TE01020415, and the BUT grant FIT-S-14-2299.

References

[1] Baeza-Yates R, Ribeiro-Neto B. Modern Information Retrieval: The Concepts and Technology behind Search. Addison-Wesley, Boston, 2011. SBN-13:978-0321416919, 10:0321416910.

[2] Buettcher S, Clarke CLA, Cormack GV. Information Retrieval: Implementing and Evaluating Search Engines. The MIT Press, Cambridge, 2010. ISBN-10:0262528878, 13:978-0262528870.

[3] Manning CD, Raghavan P, Schütze H. Introduction to Information Retrieval. Cambridge University Press, New York, 2008. ISBN: 9780521865715.

[4] Meduna A, Zemek P. Jumping Finite Automata. International Journal of Foundations of Computer Science, 2012;23(7):1555–1578. doi:10.1142/S0129054112500244.

[5] Křivka Z, Meduna A. Jumping Grammars. International Journal of Foundations of Computer Science, 2015;26(6):709–731. URL `http://dx.doi.org/10.1142/S0129054115500409`.

[6] Meduna A, Techet J. Scattered Context Grammars and their Applications. WIT Press, Southampton, 2010. ISBN:1845644263, 9781845644260.

[7] Greibach S, Hopcroft J. Scattered context grammars. Journal of Computer and System Sciences, 1969; 3(3):233–247. URL `http://dx.doi.org/10.1016/S0022-0000(69)80015-2`.

[8] Salomaa A. Formal Languages. Academic Press, London, 1973.

[9] Rozenberg G, Salomaa A (eds.). Handbook of Formal Languages, Vol. 1: Word, Language, Grammar. Springer, New York, 1997. doi:10.1007/978-3-642-59136-5.

[10] Meduna A. Automata and Languages: Theory and Applications. Springer, London, 2000. ISBN-10: 1852330740, 13:978-1852330743.

[11] Meduna A. Formal Languages and Computation. Taylor & Francis Informa plc, New York, 2014. ISBN-10:1466513454, 13:978-1466513457.

[12] Meduna A. Generative Power of Three-Nonterminal Scattered Context Grammars. Theoretical Computer Science, 2000;(246):279–284. URL `http://dx.doi.org/10.1016/S0304-3975(00)00153-5`.

[13] Harrison MA. Introduction to Formal Language Theory. Addison-Wesley Longman Publishing Co., Inc., Boston, 1978. ISBN-10:0201029553, 13:978-0201029550.

[14] Fernau H, Meduna A. A simultaneous reduction of several measures of descriptional complexity in scattered context grammars. Information Processing Letters, 2003;86(5):235–240. URL `http://dx.doi.org/10.1016/S0020-0190(03)00202-3`.

[15] Wood D. Theory of computation. Harper & Row, New York, 1987. ISBN: 0060472081.

[16] Meduna A, Zemek P. Regulated Grammars and Their Transformations. Brno University of Technology, Brno, 2010. ISBN: 978-80-214-4203-0.